

日本国特許庁
PATENT OFFICE
JAPANESE GOVERNMENT

W. Meredith
3/24/97
Jc972 U.S. PTO
09/759243
01/16/01

別紙添付の書類に記載されている事項は下記の出願書類に記載されて
いる事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed
with this Office.

出願年月日
Date of Application:

2000年 7月12日

出願番号
Application Number:

特願2000-211446

出願人
Applicant(s):

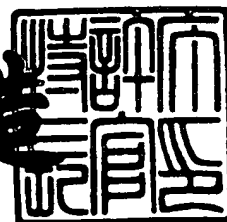
三菱電機株式会社

CERTIFIED COPY OF
PRIORITY DOCUMENT

2000年 8月 4日

特許庁長官
Commissioner,
Patent Office

及川耕造



出証番号 出証特2000-3061176

【書類名】 特許願

【整理番号】 525496JP01

【提出日】 平成12年 7月12日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 13/00

【発明者】

【住所又は居所】 東京都千代田区丸の内二丁目2番3号 三菱電機株式会社
社内

【氏名】 前川 隆昭

【発明者】

【住所又は居所】 東京都千代田区丸の内二丁目2番3号 三菱電機株式会社
社内

【氏名】 西岡 篤史

【特許出願人】

【識別番号】 000006013

【氏名又は名称】 三菱電機株式会社

【代理人】

【識別番号】 100057874

【弁理士】

【氏名又は名称】 曾我 道照

【選任した代理人】

【識別番号】 100110423

【弁理士】

【氏名又は名称】 曾我 道治

【選任した代理人】

【識別番号】 100071629

【弁理士】

【氏名又は名称】 池谷 豊

【選任した代理人】

【識別番号】 100084010

【弁理士】

【氏名又は名称】 古川 秀利

【選任した代理人】

【識別番号】 100094695

【弁理士】

【氏名又は名称】 鈴木 憲七

【選任した代理人】

【識別番号】 100081916

【弁理士】

【氏名又は名称】 長谷 正久

【選任した代理人】

【識別番号】 100087985

【弁理士】

【氏名又は名称】 福井 宏司

【手数料の表示】

【予納台帳番号】 000181

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 ソフトウェア管理システム

【特許請求の範囲】

【請求項 1】 センターサーバと、広域ネットワークを介して前記センターサーバに接続されたローカルサーバとを備えたネットワークシステムからなるソフトウェア管理システムにおいて、

前記センターサーバは、

前記ローカルサーバにダウンロードさせて動作するアプリケーションと、

前記アプリケーションの動作を記述したスクリプトと、

障害発生時の対応を行う障害対策手段とを備え、

前記ローカルサーバは、

ネットワーク指向言語実行環境と、

前記アプリケーションの前記センターサーバからのダウンロードや処理終了後の削除などを行うリモート管理手段と、

前記スクリプトを解釈して前記アプリケーションに処理を要求するスクリプト解釈手段と、

前記アプリケーションの実行中に発生するイベント情報の記録や、前記障害発生時の情報管理および復旧処理を行う高信頼性手段と

を備えたことを特徴とするソフトウェア管理システム。

【請求項 2】 前記リモート管理手段は、

センターサーバ情報およびアプリケーション情報と、

前記アプリケーションの実行要求に対応する要求処理手段と、

前記センターサーバ情報および前記アプリケーション情報に基づいて前記アプリケーションを前記センターサーバからダウンロードするアプリケーションダウンロード手段と、

前記アプリケーション情報に基づいて前記アプリケーションの起動や削除などの処理を行うアプリケーション管理手段と

を含むことを特徴とする請求項 1 に記載のソフトウェア管理システム。

【請求項 3】 前記スクリプト解釈手段は、

スクリプト定義およびイベント一覧と、

前記スクリプト定義にしたがって前記スクリプトを解釈して前記スクリプトの定義内容に相当するイベントを出力する解釈手段と、

前記イベントを取り込むとともに前記イベント一覧にしたがって前記イベントにより駆動される処理を抽出するイベント駆動手段と

を含むことを特徴とする請求項 1 に記載のソフトウェア管理システム。

【請求項 4】 前記スクリプトは、XML (eXtensible Markup Language) により記述され、

前記スクリプト定義は、DTD (Document Type Definition) により記述されたことを特徴とする請求項 1 に記載のソフトウェア管理システム。

【請求項 5】 前記障害対策手段は、

前記障害発生時の障害情報を前記ローカルサーバから取得する障害情報取得手段と、

前記障害発生時の障害対策を決定して前記ローカルサーバに通知する障害対策通知手段とを含み、

前記高信頼性手段は、

前記障害発生時に障害の発生を検知する障害検知手段と、

前記障害発生時の障害情報を収集する障害情報収集手段と、

前記障害情報を前記センターサーバに通知する障害通知手段と、

前記センターサーバからの障害対策に応じて前記障害を復旧させる復旧手段と

前記イベント情報を収集記録するイベント収集手段と

を含むことを特徴とする請求項 1 に記載のソフトウェア管理システム。

【請求項 6】 前記障害対策手段は、前記障害情報の種別毎に障害対策を格納する障害対策一覧表を含み、

前記障害対策通知手段は、

前記障害情報の種別に対応した障害対策を前記障害対策一覧表に基づいて検出する障害対策検出手段と、

前記障害対策を前記ローカルサーバに通知する通知手段と
を含むことを特徴とする請求項 5 に記載のソフトウェア管理システム。

【請求項 7】 前記障害対策手段は、前記障害発生時の障害情報を前記ローカルサーバから取得する障害情報取得手段を含み、

前記高信頼性手段は、

前記障害発生時に障害の発生を検知する障害検知手段と、

前記障害発生時の障害情報を収集する障害情報収集手段と、

前記障害発生時の障害への対応を自律的に行い前記障害を自動復旧させる復旧手段と、

前記障害情報および自動復旧の情報を前記センターサーバに通知する通知手段と、

前記イベント情報を収集記録するイベント収集手段と

を含むことを特徴とする請求項 1 に記載のソフトウェア管理システム。

【請求項 8】 前記ローカルサーバは複数存在し、それぞれ、前記ネットワーク指向言語実行環境、前記リモート管理手段、前記スクリプト解釈手段および前記高信頼性手段を備えたことを特徴とする請求項 1 に記載のソフトウェア管理システム。

【請求項 9】 前記センターサーバは複数存在し、それぞれ、前記アプリケーション、前記スクリプトおよび前記障害対策手段を備えたことを特徴とする請求項 8 に記載のソフトウェア管理システム。

【請求項 10】 前記ローカルサーバは複数存在し、

少なくとも 1 つのローカルサーバは、前記ネットワーク指向言語実行環境、前記リモート管理手段、前記スクリプト解釈手段および前記高信頼性手段を備え、

他のローカルサーバは、前記ネットワーク指向言語実行環境、前記リモート管理手段および前記スクリプト解釈手段を備えたことを特徴とする請求項 1 に記載のソフトウェア管理システム。

【請求項 11】 センターサーバと、広域ネットワークを介して前記センターサーバに接続されたローカルサーバとを備えたネットワークシステムからなるソフトウェア管理システムにおいて、

前記センターサーバは、

前記ローカルサーバにダウンロードさせて動作するアプリケーションと、

前記アプリケーションの動作を記述したスクリプトとを備え、

前記ローカルサーバは、

ネットワーク指向言語実行環境と、

前記アプリケーションの前記センターサーバからのダウンロードや処理終了後の削除などを行うリモート管理手段と、

前記スクリプトを解釈して前記アプリケーションに処理を要求するスクリプト解釈手段と

を備えたことを特徴とするソフトウェア管理システム。

【請求項 1 2】 前記ローカルサーバは複数存在し、それぞれ、前記ネットワーク指向言語実行環境、前記リモート管理手段および前記スクリプト解釈手段を備えたことを特徴とする請求項 1 1 に記載のソフトウェア管理システム。

【請求項 1 3】 センターサーバと、広域ネットワークを介して前記センターサーバに接続されたローカルサーバとを備えたネットワークシステムからなるソフトウェア管理システムにおいて、

前記センターサーバは、

前記ローカルサーバにダウンロードさせて動作するアプリケーションと、

障害発生時の対応を行う障害対策手段とを備え、

前記ローカルサーバは、

ネットワーク指向言語実行環境と、

前記アプリケーションの前記センターサーバからのダウンロードや処理終了後の削除などを行うリモート管理手段と、

前記アプリケーションの実行中に発生するイベント情報の記録や、前記障害発生時の情報管理および復旧処理を行う高信頼性手段と

を備えたことを特徴とするソフトウェア管理システム。

【請求項 1 4】 前記ローカルサーバは複数存在し、それぞれ、前記ネットワーク指向言語実行環境、前記リモート管理手段および前記高信頼性手段を備えたことを特徴とする請求項 1 3 に記載のソフトウェア管理システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

この発明は、センターサーバと、広域ネットワークを介してセンターサーバに接続されたローカルサーバとを備え、センターサーバが広域ネットワーク全体を管理するように構成されたネットワークシステムからなるソフトウェア管理システムに関し、特にセンターサーバからローカルサーバへのアプリケーションソフトウェアのダウンロード機能お信頼性を向上させたソフトウェア管理システムに関するものである。

【0002】

【従来の技術】

近年、インターネットなどの広域ネットワークの利用がますます進みつつあり、また、ネットワーク指向のプログラム言語である「J a v a」の普及にともない、広域ネットワークを介してセンターサーバからローカルサーバ（コンピュータなど）にソフトウェアをダウンロードする処理が増えてきている。

【0003】

しかも、最近では、組込み型のローカルサーバのように、メモリ容量の比較的小さいコンピュータにおいても、アプリケーションソフトウェアをダウンロードする仕組みが必要になってきている。

【0004】

しかしながら、メモリ容量の少ない組込み型のローカルサーバにおいては、全てのアプリケーションを常時メモリ上に置くことができない場合もあり、また、通常、ディスクなどの補助記憶装置も装備されていないので、必要に応じてソフトウェアをダウンロードし、不要になれば削除する機能が必要となる。

【0005】

従来より、上記のようなりモートでのソフトウェア管理機能は、たとえばOG Si (Open Gateway Service Initiative) が定めたOSG (Open Gateway Service) などの規格がある（詳細は、OG Siのホームページ「<http://www.osgi.org>」

／」を参照)。

【0006】

上記OSG規格は、ソフトウェアのダウンロードや削除の方式を規定しているが、さらに高い信頼性が必要なシステムに適用する場合には、独自に、高信頼性機能や安全確実な動作を保証するための仕組みを実装する必要がある。

【0007】

たとえば特開平11-65968号公報に記載された「ネットワーク管理方式およびそのシステム」においては、ITU-Tの勧告「M. 3010」で規定されたマネージャ・エージェントモデルに基づく方式に、Javaで記述されたソフトウェアをダウンロードする仕組みを加えたシステムが提案されている。

【0008】

上記公報に記載のネットワーク管理方式は、ITU-Tの勧告「M. 3010」で規定されているマネージャ・エージェントモデルに基づく方式を前提としているが、Javaで記述されたソフトウェア自体の動作が異常となった場合の処置や、信頼性を高めるための工夫については何ら考慮していない。

【0009】

すなわち、従来のネットワークを介したソフトウェアのリモート管理システムにおいては、いずれも、ダウンロードや削除などの基本的な機能については規定しているものの、ダウンロードされたソフトウェアが異常な動作をしたときの処理や、安全確実な動作を保証する仕組みなど、システムとしての信頼性を高める機能については何ら考慮されていない。

【0010】

今後、ネットワークを介したソフトウェアの流通がますます進むことを考慮すると、ダウンロードしたソフトウェアの安全性や、万一の異常動作時に対処することのニーズが強いことは言うまでもない。しかしながら、従来システムでは、十分にその要求に応えることはできていない。

【0011】

【発明が解決しようとする課題】

従来のソフトウェア管理システムは以上のように、ダウンロードしたソフトウ

エアの安全性を確保することや、万一の異常動作時の対処することなどの要求に応えることができないという問題点があった。

【 0 0 1 2 】

この発明は上記のような問題点を解決するためになされたもので、ネットワークを介してダウンロードされたソフトウェアの安全確実な動作を保証するとともに、万一異常が発生しても障害情報収集や復旧を支援する機能を提供し、広域ネットワーク上を流通するソフトウェアの処理の信頼性を向上させたソフトウェア管理システムを得ることを目的とする。

【 0 0 1 3 】

【課題を解決するための手段】

この発明の請求項 1 に係るソフトウェア管理システムは、センターサーバと、広域ネットワークを介してセンターサーバに接続されたローカルサーバとを備えたネットワークシステムからなるソフトウェア管理システムにおいて、センターサーバは、ローカルサーバにダウンロードさせて動作するアプリケーションと、アプリケーションの動作を記述したスクリプトと、障害発生時の対応を行う障害対策手段とを備え、ローカルサーバは、ネットワーク指向言語実行環境と、アプリケーションのセンターサーバからのダウンロードや処理終了後の削除などを行うリモート管理手段と、スクリプトを解釈してアプリケーションに処理を要求するスクリプト解釈手段と、アプリケーションの実行中に発生するイベント情報の記録や、障害発生時の情報管理および復旧処理を行う高信頼性手段とを備えたものである。

【 0 0 1 4 】

また、この発明の請求項 2 に係るソフトウェア管理システムは、請求項 1 において、リモート管理手段は、センターサーバ情報およびアプリケーション情報と、アプリケーションの実行要求に対応する要求処理手段と、センターサーバ情報およびアプリケーション情報に基づいてアプリケーションをセンターサーバからダウンロードするアプリケーションダウンロード手段と、アプリケーション情報に基づいてアプリケーションの起動や削除などの処理を行うアプリケーション管理手段とを含むものである。

【 0 0 1 5 】

また、この発明の請求項 3 に係るソフトウェア管理システムは、請求項 1 において、スクリプト解釈手段は、スクリプト定義およびイベント一覧と、スクリプト定義にしたがってスクリプトを解釈してスクリプトの定義内容に相当するイベントを出力する解釈手段と、イベントを取り込むとともにイベント一覧にしたがってイベントにより駆動される処理を抽出するイベント駆動手段とを含むものである。

【 0 0 1 6 】

また、この発明の請求項 4 に係るソフトウェア管理システムは、請求項 1 において、スクリプトは、XML (eXtensible Markup Language) により記述され、スクリプト定義は、DTD (Document Type Definition) により記述されたものである。

【 0 0 1 7 】

また、この発明の請求項 5 に係るソフトウェア管理システムは、請求項 1 において、障害対策手段は、障害発生時の障害情報をローカルサーバから取得する障害情報取得手段と、障害発生時の障害対策を決定してローカルサーバに通知する障害対策通知手段とを含み、高信頼性手段は、障害発生時に障害の発生を検知する障害検知手段と、障害発生時の障害情報を収集する障害情報収集手段と、障害情報をセンターサーバに通知する障害通知手段と、センターサーバからの障害対策に応じて障害を復旧させる復旧手段と、イベント情報を収集記録するイベント収集手段とを含むものである。

【 0 0 1 8 】

また、この発明の請求項 6 に係るソフトウェア管理システムは、請求項 5 において、障害対策手段は、障害情報の種別毎に障害対策を格納する障害対策一覧表を含み、障害対策通知手段は、障害情報の種別に対応した障害対策を障害対策一覧表に基づいて検出する障害対策検出手段と、障害対策をローカルサーバに通知する通知手段とを含むものである。

【 0 0 1 9 】

また、この発明の請求項 7 に係るソフトウェア管理システムは、請求項 1 にお

いて、障害対策手段は、障害発生時の障害情報をローカルサーバから取得する障害情報取得手段を含み、高信頼性手段は、障害発生時に障害の発生を検知する障害検知手段と、障害発生時の障害情報を収集する障害情報収集手段と、障害発生時の障害への対応を自律的に行い障害を自動復旧させる復旧手段と、障害情報および自動復旧の情報をセンターサーバに通知する通知手段と、イベント情報を収集記録するイベント収集手段とを含むものである。

【 0 0 2 0 】

また、この発明の請求項 8 に係るソフトウェア管理システムは、請求項 1 において、ローカルサーバは複数存在し、それぞれ、ネットワーク指向言語実行環境、リモート管理手段、スクリプト解釈手段および高信頼性手段を備えたものである。

【 0 0 2 1 】

また、この発明の請求項 9 に係るソフトウェア管理システムは、請求項 8 において、センターサーバは複数存在し、それぞれ、アプリケーション、スクリプトおよび障害対策手段を備えたものである。

【 0 0 2 2 】

また、この発明の請求項 1 0 に係るソフトウェア管理システムは、請求項 1 において、ローカルサーバは複数存在し、少なくとも 1 つのローカルサーバは、ネットワーク指向言語実行環境、リモート管理手段、スクリプト解釈手段および高信頼性手段を備え、他のローカルサーバは、ネットワーク指向言語実行環境、リモート管理手段およびスクリプト解釈手段を備えたものである。

【 0 0 2 3 】

また、この発明の請求項 1 1 に係るソフトウェア管理システムは、センターサーバと、広域ネットワークを介してセンターサーバに接続されたローカルサーバとを備えたネットワークシステムからなるソフトウェア管理システムにおいて、センターサーバは、ローカルサーバにダウンロードさせて動作するアプリケーションと、アプリケーションの動作を記述したスクリプトとを備え、ローカルサーバは、ネットワーク指向言語実行環境と、アプリケーションのセンターサーバからのダウンロードや処理終了後の削除などを行うリモート管理手段と、スクリプ

トを解釈してアプリケーションに処理を要求するスクリプト解釈手段とを備えたものである。

【 0 0 2 4 】

また、この発明の請求項 1 2 に係るソフトウェア管理システムは、請求項 1 1 において、ローカルサーバは複数存在し、それぞれ、ネットワーク指向言語実行環境、リモート管理手段およびスクリプト解釈手段を備えたものである。

【 0 0 2 5 】

また、この発明の請求項 1 3 に係るソフトウェア管理システムは、センターサーバと、広域ネットワークを介してセンターサーバに接続されたローカルサーバとを備えたネットワークシステムからなるソフトウェア管理システムにおいて、センターサーバは、ローカルサーバにダウンロードさせて動作するアプリケーションと、障害発生時の対応を行う障害対策手段とを備え、ローカルサーバは、ネットワーク指向言語実行環境と、アプリケーションのセンターサーバからのダウンロードや処理終了後の削除などを行うリモート管理手段と、アプリケーションの実行中に発生するイベント情報の記録や、障害発生時の情報管理および復旧処理を行う高信頼性手段とを備えたものである。

【 0 0 2 6 】

また、この発明の請求項 1 4 に係るソフトウェア管理システムは、請求項 1 3 において、ローカルサーバは複数存在し、それぞれ、ネットワーク指向言語実行環境、リモート管理手段および高信頼性手段を備えたものである。

【 0 0 2 7 】

【発明の実施の形態】

実施の形態 1.

以下、図面を参照しながら、この発明の実施の形態 1 について詳細に説明する。

図 1 はこの発明の実施の形態 1 を概念的に示すブロック構成図である。

【 0 0 2 8 】

図 1 において、センターサーバ 1 は、アプリケーション 1 1 と、アプリケーションの動作を記述したスクリプト 1 2 と、障害発生時の対応を行う障害対策手段

1 3 とを有する。

【 0 0 2 9 】

アプリケーション 1 1 は、後述するように、複数のアプリケーション 1 1 a ~ 1 1 x を含み、スクリプト 1 2 は、各アプリケーション 1 1 a ~ 1 1 x に対応した複数のスクリプト 1 2 a ~ 1 2 x を含む。

【 0 0 3 0 】

広域ネットワーク 2 は、センターサーバ 1 とローカルサーバ 3 とを接続している。

センターサーバ 1 内のアプリケーション 1 1 は、広域ネットワーク 2 を介してローカルサーバ 3 にダウンロードされて動作する。

【 0 0 3 1 】

ローカルサーバ 3 は、必要に応じてセンターサーバ 1 からダウンロードされるアプリケーション 1 1 a ~ 1 1 x と、J a v a V M からなるネットワーク指向言語実行環境 3 1 と、リモート管理手段 3 2 と、スクリプト解釈手段 3 3 と、高信頼性手段 3 4 とを有する。

【 0 0 3 2 】

リモート管理手段 3 2 は、アプリケーション 1 1 のセンターサーバ 1 からのダウンロードや処理終了後の削除などの処理を実行する。

高信頼性手段 3 4 は、アプリケーション 1 1 の実行中に発生するイベント情報の記録や、障害発生時の情報管理および復旧処理を行う。

【 0 0 3 3 】

ローカルネットワーク 4 は、ローカルサーバ 3 と管理制御対象機器 5 とを接続している。

【 0 0 3 4 】

次に、図 1 に示したこの発明の実施の形態 1 によるシステム全体の動作について説明する。

まず、センターサーバ 1 内にあらかじめ登録されているアプリケーション 1 1 のうち、ローカルサーバ 3 に必要なアプリケーション 1 1 a ~ 1 1 x は、広域ネットワーク 2 を経由してローカルサーバ 3 にダウンロードされる。

【 0 0 3 5 】

ダウンロードされたアプリケーション 1 1 a ~ 1 1 x は、ローカルネットワーク 4 を介して接続された管理制御対象機器 5 に対し、機器の制御や機器内の情報の取得などの処理を行う。

【 0 0 3 6 】

次に、図 2 のフローチャートを参照しながら、図 1 に示したこの発明の実施の形態 1 による具体的な動作について説明する。

すなわち、ダウンロードされたソフトウェアの安全で確実な動作の仕組みと、万一の異常発生時の対応の仕組みとについて説明する。

【 0 0 3 7 】

なお、ここで、センターサーバ 1 からダウンロードして動作するネットワーク指向のソフトウェア実行環境 3 1 には、現在もっとも普及している「J a v a」の実行環境「J a v a V M (J a v a V i r t u a l M a c h i n e)」を想定している。

【 0 0 3 8 】

ただし、同種の機能を持つ環境であれば、「J a v a V M」以外のものでも、同等の効果を奏することは言うまでもない。

【 0 0 3 9 】

図 2 において、まず、ローカルサーバ 3 内のリモート管理手段 3 2 により、センターサーバ 1 からアプリケーション 1 1 をローカルサーバ 3 にダウンロードする（ステップ S 1）。

【 0 0 4 0 】

ダウンロードされたアプリケーション 1 1 は、スクリプト解釈手段 3 3 から処理要求が発生するまでは、待機状態になる（ステップ S 2）。

【 0 0 4 1 】

ここで、アプリケーション 1 1 に対する処理要求が発生し、ステップ S 2 において処理要求有り（すなわち、Y E S）と判定され、この処理要求がリモート管理手段 3 2 に入力されると、この時点からアプリケーション 1 1 に起動がかかる（ステップ S 3）。

【 0 0 4 2 】

このとき、処理要求に対応したスクリプト 1 2 は、センターサーバ 1 からローカルサーバ 3 にダウンロードされ、ローカルサーバ 3 内のスクリプト解釈手段 3 3 によって、その処理内容が解釈される。

【 0 0 4 3 】

また、スクリプト解釈手段 3 3 は、解釈された処理内容にしたがう処理要求をアプリケーション 1 1 に送り、処理を実行させる（ステップ S 4）。

このように、アプリケーション 1 1 を実行させるためにスクリプト 1 2 を仲介させるのは、安全で確実な動作を実行させるためである。

【 0 0 4 4 】

すなわち、アプリケーション 1 1 は、センターサーバ 1 への登録時に動作検証が実行されるが、このとき、アプリケーション 1 1 の動作をスクリプト 1 2 により規定しておけば、動作範囲が決定し、スクリプト 1 2 に記載されている範囲外の動作は実行されない。

【 0 0 4 5 】

また、あらかじめ、全ての動作範囲をスクリプト 1 2 で規定しておけば、動作検証はスクリプト 1 2 に記載の範囲のみで実行すればよく、効率的で確実な検証を実行することができる。

【 0 0 4 6 】

したがって、あらかじめ、全ての動作の検証を実行することができ、アプリケーション 1 1 の実行時の安全性を保証することができる。

また、スクリプト 1 2 に記載された動作のみが実行されるので、仕様が明確になり、確実に必要な動作を確認することができる。

【 0 0 4 7 】

処理要求が発生したアプリケーション 1 1 の動作中においては、アプリケーション 1 1 自体から、または、OS などのシステムから、種々のイベントが発行される。

【 0 0 4 8 】

図 2 に戻り、高信頼性手段 3 4 は、発行されたイベントを検知し、イベント情

報として内部に記録する（ステップ S 5）。

【0049】

続いて、高信頼性手段 3 4 は、発行されたイベントに関して障害が発生したか否かを判定し（ステップ S 6）、障害発生のイベントである（すなわち、YES）と判定された場合には、イベント情報の蓄積処理（ステップ S 5）のみでなく、所定の手続きにしたがって障害への対応処理を実行し（ステップ S 7）、ステップ S 8 に進む。

【0050】

また、ステップ S 6 において、イベントに障害がない（すなわち、NO）と判定されれば、直ちにステップ S 8 に進む。

ステップ S 8 においては、全ての処理が終了したか否かを判定し、全ての処理が終了した（すなわち、YES）と判定されれば、図 2 の処理ルーチンを終了して、次のアプリケーション 1 1 への処理要求を待つ。

【0051】

一方、ステップ S 8 において、全ての処理が終了していない（すなわち、NO）と判定されれば、イベント情報の記録ステップ S 5 に戻り、上記処理を繰り返す。

【0052】

このように、高信頼性手段 3 4 は、動作中のアプリケーション 1 1 が発行するイベント情報を記録すること（ステップ S 5）により、アプリケーション 1 1 の異常を検知して必要な対応処置（ステップ S 7）を実行する。

この機能により、動作中に万一異常が起こっても適切な対応処置が行われ、システムとしての機能が損なわれることはない。

【0053】

実施の形態 2.

なお、上記実施の形態 1 では、ローカルサーバ 3 内のリモート管理手段 3 2 について詳述しなかったが、リモート管理手段 3 2 を図 3 のように構成してもよい。

【0054】

以下、図 3 を参照しながら、リモート管理手段 3 2 を具体化したこの発明の実施の形態 2 について説明する。

【 0 0 5 5 】

図 3 はローカルサーバ 3 内のリモート管理手段 3 2 の機能を具体的に示すブロック構成図であり、前述（図 1 参照）と同様のものについては、同一符号を付して詳述を省略する。

【 0 0 5 6 】

図 3 において、ローカルネットワーク 4 および管理制御対象機器 5 などの、リモート管理手段 3 2 と直接関連しない構成は省略されている。

また、センターサーバ 1 においては、スクリプト 1 2 a およびアプリケーション 1 1 a のみが代表的に示されている。

【 0 0 5 7 】

リモート管理手段 3 2 は、センターサーバ情報 3 2 1 と、アプリケーション情報 3 2 2 と、要求処理手段 3 2 3 と、アプリケーションダウンロード手段 3 2 4 と、アプリケーション管理手段 3 2 5 とを備えている。

【 0 0 5 8 】

要求処理手段 3 2 3 は、スクリプト解釈手段 3 3 から発生されたアプリケーション 1 1 の実行要求に対応する。

【 0 0 5 9 】

アプリケーションダウンロード手段 3 2 4 は、センターサーバ情報 3 2 1 およびアプリケーション情報 3 2 2 に基づいて、アプリケーション 1 1 をセンターサーバ 1 からダウンロードする。

【 0 0 6 0 】

アプリケーション管理手段 3 2 5 は、アプリケーション情報 3 2 2 に基づいて、アプリケーション 1 1 の起動や削除などの処理を実行する。

【 0 0 6 1 】

次に、図 4 のフローチャートを参照しながら、図 3 に示したこの発明の実施の形態 2 による具体的な動作について説明する。

なお、システムとしての動作は前述と同様なので、ここでは、リモート管理手

段 3 2 内の処理動作のみに注目して説明する。

【 0 0 6 2 】

まず、スクリプト解釈手段 3 3 からアプリケーション 1 1 a に対する起動要求が発生すると、リモート管理手段 3 2 内の要求処理手段 3 2 3 に、アプリケーション 1 1 a に対する起動要求が入力される（ステップ S 1 1）。

【 0 0 6 3 】

続いて、起動対象となるアプリケーション 1 1 a が存在するか否かを判定し（ステップ S 1 2）、ローカルサーバ 3 内にアプリケーション 1 1 a が存在しない（すなわち、N O）と判定されれば、アプリケーションダウンロード手段 3 2 4 は、要求処理手段 3 2 3 からの依頼を受け、センターサーバ情報 3 2 1 に基づいてセンターサーバ 1 にアクセスする（ステップ S 1 3）。

【 0 0 6 4 】

このように、アプリケーション情報 3 2 2 に基づいて、センターサーバ 1 から起動対象のアプリケーション 1 1 a をダウンロードする（ステップ S 1 4）。

なお、アプリケーション情報 3 2 2 は、ローカルサーバ 3 内でなくセンターサーバ 1 内にあってもよい。

【 0 0 6 5 】

こうして、アプリケーション 1 1 a がローカルサーバ 3 内に存在する状態になると、要求処理手段 3 2 3 は、アプリケーション 1 1 a の起動をかける（ステップ S 1 5）。

【 0 0 6 6 】

なお、ステップ S 1 2 において、ローカルサーバ 3 内に起動対象のアプリケーション 1 1 a が存在する（すなわち、Y E S）と判定されれば、直ちにステップ S 1 5 に進む。

【 0 0 6 7 】

また、ステップ S 1 5 に続いて、アプリケーション 1 1 a の処理を記述したスクリプト 1 2 a をセンターサーバ 1 から読み込み、このスクリプト 1 2 a をスクリプト解釈手段 3 3 に送付して、アプリケーション 1 1 a に処理を実行させる（ステップ S 1 6）。

【 0 0 6 8 】

実際のアプリケーション 1 1 a の動作は、スクリプト 1 2 a の内容にしたがって実行される。

【 0 0 6 9 】

次に、アプリケーション 1 1 a の処理が完了したか否かを判定し（ステップ S 1 7）、処理が完了していない（すなわち、N O）と判定されれば、ステップ S 1 6 の処理を繰り返す。

【 0 0 7 0 】

また、ステップ S 1 7 において、アプリケーション 1 1 a の処理が完了した（すなわち、Y E S）と判定されれば、続いて、アプリケーション 1 1 a が常駐型か否（非常駐型）かを判定する（ステップ S 1 8）。

【 0 0 7 1 】

ステップ S 1 8 において、アプリケーション 1 1 a が非常駐型である（すなわち、N O）と判定されれば、アプリケーション 1 1 a を削除し（ステップ S 1 9）、図 4 の処理ルーチンを終了する。

【 0 0 7 2 】

また、ステップ S 1 8 において、アプリケーション 1 1 a が常駐型である（すなわち、Y E S）と判定されれば、直ちに図 4 の処理ルーチンを終了し、アプリケーション 1 1 a をそのままメモリ内にとどめておく。

【 0 0 7 3 】

以上のように、リモート管理手段 3 2 において、効率的なソフトウェアのダウンロードが実行可能になる。

また、非常駐型のアプリケーション 1 1 を動作終了毎に削除することにより、メモリ容量の少ないローカルサーバ 3 においても、処理の制約を受けずにシステムの運用を実行することができる。

【 0 0 7 4 】

実施の形態 3.

なお、上記実施の形態 1 では、ローカルサーバ 3 内のスクリプト解釈手段 3 3 について詳述しなかったが、スクリプト解釈手段 3 3 を図 5 ～ 図 7 のように構成

してもよい。

【 0 0 7 5 】

以下、図 5 ～ 図 7 を参照しながら、スクリプト解釈手段 3 3 を具体化したこの発明の実施の形態 3 について説明する。

図 5 はローカルサーバ 3 内のスクリプト解釈手段 3 3 の機能を具体的に示すブロック構成図であり、スクリプト解釈手段 3 3 の周辺構成のみが示されている。

【 0 0 7 6 】

なお、図 5 に示されない構成は、前述（図 1、図 3 参照）と同様である。

図 5 において、スクリプト解釈手段 3 3 は、スクリプト定義 3 3 1、イベント一覧 3 3 2、解釈手段 3 3 3 およびイベント駆動手段 3 3 4 を備えている。

【 0 0 7 7 】

解釈手段 3 3 3 は、スクリプト定義 3 3 1 を含み、スクリプト定義 3 3 1 にしたがってスクリプト 1 2 を解釈し、スクリプト 1 2 の定義内容に相当するイベントを出力する。

【 0 0 7 8 】

イベント駆動手段 3 3 4 は、イベント一覧 3 3 2 を含み、イベントを取り込むとともに、イベント一覧 3 3 2 にしたがって、イベントにより駆動される処理を抽出する。

【 0 0 7 9 】

図 6 はこの発明の実施の形態 3 によるスクリプト定義 3 3 1 の一例を示す説明図であり、図 7 はこの発明の実施の形態 3 によるイベント一覧 3 3 2 の一例を示す説明図である。

【 0 0 8 0 】

図 6 において、スクリプト定義 3 3 1 は、スクリプト 1 2 a 内の各定義内容に相当するイベントをマップ形式で格納しており、たとえば、定義内容<スクリプト a >に相当するイベント A が示されている。

【 0 0 8 1 】

図 7 において、イベント一覧 3 3 2 は、各イベントに相当するアプリケーションおよび処理部をマップ形式で格納しており、たとえば、イベント A に相当する

アプリケーション 1 1 a および処理部 A が示されている。

【 0 0 8 2 】

次に、図 8 のフローチャートを参照しながら、図 5 ～図 7 に示したこの発明の実施の形態 3 による動作について説明する。

ここでは、スクリプト解釈手段 3 3 の処理のみに注目して説明する。

【 0 0 8 3 】

まず、ローカルサーバ 3 は、リモート管理手段 3 2 （図 1、図 3 参照）の介在により、センターサーバ 1 内のスクリプト 1 2 a をスクリプト解釈手段 3 3 内の解釈手段 3 3 3 に読み込む（ステップ S 2 1）。

【 0 0 8 4 】

次に、解釈手段 3 3 3 は、スクリプト定義 3 3 1 にしたがって、定義内容＜スクリプト a＞に相当するイベント A を抽出する（ステップ S 2 2）。

【 0 0 8 5 】

続いて、イベント駆動手段 3 3 4 は、イベント一覧 3 3 2 にしたがって、イベント A に対応する処理（アプリケーション 1 1 a、処理 A）を抽出し（ステップ S 2 3）、アプリケーション 1 1 a に対して、当該処理の実行を要求する（ステップ S 2 4）。

【 0 0 8 6 】

この結果、スクリプト解釈手段 3 3 において、安全且つ確実に処理を実行するためのスクリプト 1 2 の仕様を柔軟に定義することができる。

すなわち、スクリプト定義 3 3 1 によって、処理とは独立する形式でスクリプト 1 2 を定義することができる。

【 0 0 8 7 】

また、スクリプト 1 2 自体は、システムを構成するネットワーク、ハードウェアおよびソフトウェアなどの仕様から制約を受けることはなく、純粹に処理の意味論的な定義を行うように規定することができる。

【 0 0 8 8 】

なお、スクリプト定義 3 3 1 は、上記意味論的に規定されたスクリプト 1 2 と実際のシステムとの橋渡しを行うもので、スクリプト 1 2 内の記述とシステムの

仕様に基づいたイベントとの対応を定義する。

【 0 0 8 9 】

一方、イベント一覧 3 3 2 は、システムの仕様に基づいたイベントと実際に動作するソフトウェアの実装との橋渡しを行うもので、上記構成によりソフトウェアの設計の自由度を広げることができ、或るイベントに対応したソフトウェアの構成を自由に組むことができる。

【 0 0 9 0 】

たとえば、メモリ容量の少ないローカルサーバ 3 においては、メモリ消費が少なくなる実装を行い、メモリ容量の制約よりも処理速度が要求されるようなローカルサーバ 3 においては、性能重視の実装を行うことができる。

【 0 0 9 1 】

また、上記実装の使い分けは、ローカルサーバ 3 内のイベント一覧 3 3 2 を変更することのみによって、容易に行うことができる。

【 0 0 9 2 】

実施の形態 4 .

なお、上記実施の形態 3 では、スクリプト 1 2 およびスクリプト定義 3 3 1 の記述形式について詳述しなかったが、それぞれ、図 9 および図 1 0 のように記述してもよい。

【 0 0 9 3 】

以下、図 9 および図 1 0 を参照しながら、スクリプト 1 2 およびスクリプト定義 3 3 1 の記述形式を具体化したこの発明の実施の形態 4 について説明する。

図 9 および図 1 0 はスクリプト 1 2 およびスクリプト定義 3 3 1 のサービス記述例を示す説明図である。

【 0 0 9 4 】

図 9 において、スクリプト解釈手段 3 3 (図 5 参照) 内で扱われるスクリプト 1 2 は、XML (eXtensible Markup Language) により記述されている。

【 0 0 9 5 】

図 1 0 において、解釈手段 3 3 3 内のスクリプト定義 3 3 1 は、上記 XML 文

書に対応して、DTD (Document Type Definition) により記述されている。

【0096】

図9のように、或るアプリケーション11の動作を記述するスクリプト12がXMLで与えられた場合、スクリプト解釈手段33においては、汎用のXMLパーサが使用される。

【0097】

このとき、パーサの標準仕様であるSAX (Simple API for XML) を用いれば、タグの開始終了などのイベントを抽出することができる。

【0098】

図9および図10のように記述されたエアコン制御サービスの場合、エアコン制御開始、初期化開始、エアコン初期化開始、温度設定、風速設定、エアコン初期化終了、温度センサ初期化開始、…というイベントが順次抽出される。

【0099】

また、スクリプト解釈手段33内のイベント駆動手段334は、イベント一覧332に基づいて、対応する処理を実行していく。

【0100】

以上のように、発生させるイベント毎に、XMLのタグとして定義することにより、スクリプト12 (XMLドキュメント) から標準のXMLパーサを用いてイベントを抽出することができる。

【0101】

また、解釈手段333内のスクリプト定義331を、XMLに対応したDTD形式 (図10参照) で記述することにより、XMLパーサの機能に基づいて、XMLドキュメント (スクリプト12) の正当性検証を実行することができる。

【0102】

すなわち、DTDの検証により、スクリプト12が正しく記述されていることを保証することができる。

【0103】

実施の形態5.

なお、上記実施の形態 1 では、センターサーバ 1 内の障害対策手段 1 3 およびローカルサーバ 3 内の高信頼性手段 3 4 について詳述しなかったが、障害対策手段 1 3 および高信頼性手段 3 4 を図 1 1 のように構成してもよい。

【0104】

以下、図 1 1 を参照しながら、障害対策手段 1 3 および高信頼性手段 3 4 を具体化したこの発明の実施の形態 5 について説明する。

図 1 1 は障害対策手段 1 3 および高信頼性手段 3 4 の機能を具体的に示すブロック構成図であり、図示されない構成は前述（図 1 参照）と同様である。

【0105】

図 1 1 において、障害対策手段 1 3 は、障害発生時の障害情報をローカルサーバ 3 から取得する障害情報取得手段 1 3 1 と、障害発生時の障害対策を決定してローカルサーバ 3 に障害対策を通知する障害対策通知手段 1 3 2 とを備えている。

【0106】

高信頼性手段 3 4 は、障害情報 3 4 1、イベント情報 3 4 2、障害検知手段 3 4 3、障害情報収集手段 3 4 4、障害通知手段 3 4 5、復旧手段 3 4 6 およびイベント収集手段 3 4 7 を備えている。

【0107】

障害検知手段 3 4 3 は、障害の発生を検知し、障害情報収集手段 3 4 4 は、障害発生時の障害情報 3 4 1 を収集し、障害通知手段 3 4 5 は、障害情報 3 4 1 をセンターサーバ 1 に通知する。

【0108】

復旧手段 3 4 6 は、センターサーバ 1 からの障害対策に応じて障害を復旧させる。

イベント収集手段 3 4 7 は、イベント情報 3 4 2 を収集記録する。

【0109】

次に、図 1 2 のフローチャートを参照しながら、この発明の実施の形態 5 による動作について説明する。

ここでは、センターサーバ 1 内の障害対策手段 1 3 と、ローカルサーバ 3 内の

高信頼性手段 3 4 との連携処理に注目して説明する。

【0 1 1 0】

図 1 2 において、まず、アプリケーション 1 1 またはシステム（ローカルサーバ 3 の OS など）からイベントが発生すると（ステップ S 3 1）、高信頼性手段 3 4 内のイベント収集手段 3 4 7 は、発生したイベントを収集し、このイベントが障害関連イベントか否かを判定する（ステップ S 3 2）。

【0 1 1 1】

ステップ S 3 2 において、収集されたイベントが通常イベントであって、障害関連（異常時）のイベントではない（すなわち、NO）と判定されれば、そのイベントをイベント情報 3 4 2 として記録して（ステップ S 3 3）、図 1 2 の処理ルーチンを終了する。

【0 1 1 2】

一方、ステップ S 3 2 において、収集されたイベントが障害関連（異常時）のイベントである（すなわち、YES）と判定されれば、障害が発生したことを障害検知手段 3 4 3 に通知する（ステップ S 3 4）。

【0 1 1 3】

障害検知手段 3 4 3 は、異常時のイベントを調べて障害の種類（障害情報）を特定するとともに、障害情報収集手段 3 4 4 を起動して障害情報 3 4 1 として記録する（ステップ S 3 5）。

【0 1 1 4】

こうして検知された障害情報 3 4 1 は、続いて、障害通知手段 3 4 5 を介してセンターサーバ 1 に送られる（ステップ S 3 6）。

これに応答して、センターサーバ 1 内の障害対策手段 1 3 は、以下の処理を実行する。

【0 1 1 5】

すなわち、障害情報取得手段 1 3 1 は、ローカルサーバ 3 から到着した障害情報 3 4 1 を受け、障害対策通知手段 1 3 2 は、障害情報 3 4 1 に対する適切な障害対策を決定してローカルサーバ 3 に送信する（ステップ S 3 7）。

【0 1 1 6】

これに応答して、ローカルサーバ 3 内の高信頼性手段 3 4 は、復旧手段 3 4 6 による処理を実行する。すなわち、センターサーバ 1 から送られてきた障害対策に基づいて、障害からの復旧作業を実行し（ステップ S 3 8）、図 1 2 の処理ルーチンを終了する。

【0 1 1 7】

このように、センターサーバ 1 内の障害対策手段 1 3 とローカルサーバ 3 内の高信頼手段 3 4 との間の連携処理により、異常が発生してもシステム全体の動作不全を未然に防止することができる。

【0 1 1 8】

また、ログとしてローカルサーバ 3 内に蓄えられた障害情報 3 4 1 およびイベント情報 3 4 2 は、後々のシステム解析などに利用することができ、万一システム全体が停止した場合でも、その解析をスムーズに行うことができる。

【0 1 1 9】

なお、障害情報 3 4 1 は、センターサーバ 1 に送られるので、ローカルサーバ 3 内でなく、センターサーバ 1 内で保存されてもよい。

【0 1 2 0】

実施の形態 6.

なお、上記実施の形態 5 では、障害対策手段 1 3 内の障害対策通知手段 1 3 2 について詳述しなかったが、障害対策通知手段 1 3 2 を図 1 3 のように構成してもよい。

【0 1 2 1】

以下、図 1 3 を参照しながら、障害対策手段 1 3 内の障害対策通知手段 1 3 2 を具体化したこの発明の実施の形態 6 について説明する。

図 1 3 は障害対策通知手段 1 3 2 の機能を具体的に示すブロック構成図であり、図示されない構成は前述（図 1、図 1 1 参照）と同様である。

【0 1 2 2】

図 1 3 において、センターサーバ 1 内の障害対策手段 1 3 は、前述の障害情報取得手段 1 3 1 および障害対策通知手段 1 3 2 に加えて、障害対策一覧表 1 3 3 を備えている。

【 0 1 2 3 】

障害対策一覧表 1 3 3 は、障害情報の種別と障害対策とを対応付けて、障害情報の種別毎の障害対策をマップ形式で格納している。

【 0 1 2 4 】

障害対策通知手段 1 3 2 は、障害対策検出手段 1 3 2 1 と、通知手段 1 3 2 2 とを備えている。

【 0 1 2 5 】

障害対策検出手段 1 3 2 1 は、障害情報の種別に対応した対策方式を障害対策一覧表 1 3 3 に基づいて検出する。

通知手段 1 3 2 2 は、障害対策一覧表 1 3 3 から得られた障害対策を、ローカルサーバ 3 に通知する。

【 0 1 2 6 】

次に、図 1 4 のフローチャートを参照しながら、この発明の実施の形態 6 による動作について説明する。

ここでは、センターサーバ 1 内の障害対策処理に注目して説明する。

【 0 1 2 7 】

図 1 4 において、ステップ S 3 6 および S 3 8 は前述（図 1 2 参照）と同様の処理であり、図 1 4 に示されない処理ステップ S 3 1 ～ S 3 5 は、図 1 2 と同様である。

【 0 1 2 8 】

前述のように、ローカルサーバ 3 で障害が発生すると、障害通知手段 3 4 5 を介して、センターサーバ 1 に障害情報（種別）が送られてくる（ステップ S 3 6 ）。

【 0 1 2 9 】

センターサーバ 1 内の障害情報取得手段 1 3 1 は、ローカルサーバ 3 からの障害情報を受けて（ステップ S 3 7 1）、これを障害対策通知手段 1 3 2 内の障害対策検出手段 1 3 2 1 に渡す。

【 0 1 3 0 】

障害対策検出手段 1 3 2 1 は、障害対策一覧表 1 3 3 に基づいて、送られてき

た障害情報の種別に対応する障害対策を検出する（ステップ S 3 7 2）。

【 0 1 3 1 】

続いて、対策通知手段 1 3 2 2 は、検出された障害対策をローカルサーバ 3 内の復旧手段 3 4 6 に送信する（ステップ S 3 7 3）。

これにより、復旧手段 3 4 6 は、障害対策に対応した処置を実行する（ステップ S 3 8）。

【 0 1 3 2 】

このように、センターサーバ 1 内の障害対策手段 1 3 において、障害対策一覧表 1 3 3 に基づく障害対策検知手段 1 3 2 1 を設けることにより、あらかじめ障害に対応した設計が行い易くなり、システム全体の予防保全が可能となる。

【 0 1 3 3 】

また、障害対策一覧表 1 3 3 は、更新可能なので、新たに障害情報の種別が増えた場合や、より効率的な障害対策が提案された場合にも、リアルタイムに対応することができ、直ちにシステムに反映させることができる。

【 0 1 3 4 】

実施の形態 7.

なお、上記実施の形態 5、6 では、センターサーバ 1 内の障害対策手段 1 3 に障害対策通知手段 1 3 2 を設け、障害情報の種別に応じた障害対策をローカルサーバ 3 に通知したが、ローカルサーバ 3 内の高信頼性手段 3 4 において自律的に障害対策を施してもよい。

【 0 1 3 5 】

以下、図 1 5 を参照しながら、高信頼性手段 3 4 において自律的に障害対策を施すようにしたこの発明の実施の形態 7 について説明する。

図 1 5 は高信頼性手段 3 4 の機能を具体的に示すブロック構成図であり、前述（図 1、図 1 1、図 1 3 参照）と同様のものについては、同一符号を付して詳述を省略する。

【 0 1 3 6 】

図 1 5 において、センターサーバ 1 内の障害対策手段 1 3 は、障害情報取得手段 1 3 1 を備えているものの、前述（図 1 1、図 1 3 参照）の障害対策通知手段

1 3 2 が削除された構成となっている。

【 0 1 3 7 】

ローカルサーバ 3 内の高信頼性手段 3 4 は、前述の障害情報 3 4 1、イベント情報 3 4 2、障害検知手段 3 4 3、障害情報収集手段 3 4 4 およびイベント収集手段 3 4 7 に加えて、復旧手段 3 4 8 および通知手段 3 4 9 を備えている。

【 0 1 3 8 】

復旧手段 3 4 8 は、障害発生時の障害への対応を自律的に行い、障害を自動復旧させる。

通知手段 3 4 9 は、障害情報 3 4 1 および自動復旧の情報をセンターサーバ 1 内の障害情報取得手段 1 3 1 に通知する。

【 0 1 3 9 】

次に、図 1 6 のフローチャートを参照しながら、この発明の実施の形態 7 による高信頼性手段 3 4 の動作について説明する。

図 1 2 において、ステップ S 3 1 ～ S 3 5 は前述（図 1 2 参照）と同様の処理である。

【 0 1 4 0 】

この場合、イベント発生（ステップ S 3 1）から障害情報の収集（ステップ S 3 5）までは前述と同様である。

以下、ローカルサーバ 3 内の高信頼性手段 3 4 は、センターサーバ 1 に障害情報を通知せずに、ローカルサーバ 3 内で障害に対する自動復旧処置（ステップ S 3 9）を完結する。

【 0 1 4 1 】

最後に、障害情報 3 4 1 および自動復旧情報の結果のみをセンターサーバ 1 内の障害情報取得手段 1 3 1 に通知し（ステップ S 4 0）、図 1 6 の処理ルーチンを終了する。

【 0 1 4 2 】

これにより、センターサーバ 1 内の障害対策手段 1 3 の機能構成が簡略化されるとともに、障害情報 3 4 1 や障害対策の通知処理（ステップ S 4 0）が、ローカルサーバ 3 からセンターサーバ 1 への 1 回で済み、広域ネットワーク 2 に流れ

る通信量を低減させることができる。

【 0 1 4 3 】

また、障害発生時において、ローカルサーバ 3 内で直ちに復旧が完結するので、障害への対処時間を短縮することができる。

【 0 1 4 4 】

なお、この障害対処方式は、障害の程度が非常に軽微な場合や、障害を起こしたアプリケーション 1 1 を単に再起動するのみで済む場合など、単純な処置の場合に有効である。

したがって、複雑な対処が必要となる障害への対応は、前述の実施の形態 5 および 6 で示した障害対処方式の方が有効となる場合もあり得る。

【 0 1 4 5 】

実施の形態 8.

なお、上記実施の形態 1 ～ 7 では、センターサーバ 1 に接続されるローカルサーバ 3 の数について言及しなかったが、センターサーバ 1 に複数のローカルサーバ 3 を接続してもよい。

【 0 1 4 6 】

以下、図 1 7 を参照しながら、センターサーバ 1 に複数のローカルサーバ 3 を接続したこの発明の実施の形態 8 について説明する。

図 1 8 はこの発明の実施の形態 8 を示すブロック構成図であり、前述（図 1 参照）と同様のものについては、同一符号を付して詳述を省略する。

【 0 1 4 7 】

図 1 7 において、センターサーバ 1 には、たとえば n 個のローカルサーバ 3 a ～ 3 n が接続されている。

同一構成のローカルサーバ 3 a ～ 3 n には、それぞれ、ローカルネットワーク 4 a ～ 4 n を介して管理制御対象機器 5 a ～ 5 n が接続されている。

【 0 1 4 8 】

図 1 7 のように、複数のローカルサーバ 3 a ～ 3 n が存在するシステムであっても、センターサーバ 1 は論理的に 1 台のみでよく、基本的な動作は前述と同様であり、前述と同等の作用効果を奏する。

【 0 1 4 9 】

したがって、大規模なネットワーク指向のソフトウェア流通システムにおいても、前述と同様に信頼性を向上させることができる。

【 0 1 5 0 】

なお、システムの規模によっては、システム規模の増大に対応してセンターサーバ1の台数を増やし、1つのセンターサーバ1の負荷を軽減させることもできる。

【 0 1 5 1 】

また、複数のセンターサーバ1を設けた場合、センターサーバ機能に冗長性を持たせることができるので、1つのセンターサーバがダウンしても他のセンターサーバによりバックアップすることができ、さらに信頼性を向上させることができる。

【 0 1 5 2 】

実施の形態9.

なお、上記実施の形態1では、センターサーバ1内に障害対策手段13を設け、ローカルサーバ3内に高信頼性手段34を設けたが、障害対策手段13および高信頼性手段34を省略してもよい。

【 0 1 5 3 】

以下、図18を参照しながら、障害対策手段13および高信頼性手段34を省略したこの発明の実施の形態9について説明する。

図18はこの発明の実施の形態9を示すブロック構成図であり、前述（図1参照）と同様のものについては、同一符号を付して詳述を省略する。

【 0 1 5 4 】

図18において、センターサーバ1およびローカルサーバ3は、前述（図1参照）の障害対策手段13および高信頼性手段34が削除された構成となっている。

【 0 1 5 5 】

したがって、図18に示したこの発明の実施の形態9による動作は、前述（図2参照）のフローチャートから、障害対策手段13および高信頼性手段34に関

連するステップ S 5 ～ S 8 を削除したものとなる。

【 0 1 5 6 】

この場合、ローカルサーバ 3 内で動作するソフトウェアを非常に単純化させることができる。

ただし、図 1 8 の構成は、スクリプト 1 2 による動作制御のみで十分に信頼性を確保することができる場合に有効である。

【 0 1 5 7 】

これにより、ローカルサーバ 3 のソフトウェア構成が単純化され、少ないメモリ容量のローカルサーバ 3 を用いても、信頼性の高い処理を実現することができる。

【 0 1 5 8 】

実施の形態 1 0 .

なお、上記実施の形態 9 では、センターサーバ 1 に接続されるローカルサーバ 3 の数について言及しなかったが、前述の実施の形態 8 と同様に、センターサーバ 1 に複数のローカルサーバ 3 を接続してもよい。

【 0 1 5 9 】

以下、図 1 9 を参照しながら、センターサーバ 1 に複数のローカルサーバ 3 を接続したこの発明の実施の形態 1 0 について説明する。

図 1 9 はこの発明の実施の形態 1 0 を示すブロック構成図であり、前述（図 1 8 参照）と同様のものについては、同一符号を付して詳述を省略する。

【 0 1 6 0 】

図 1 9 において、センターサーバ 1 には、複数のローカルサーバ 3 a ～ 3 n が接続されている。

図 1 9 のように、複数のローカルサーバ 3 a ～ 3 n を有するシステムにおいても、センターサーバ 1 は論理的に 1 台でよく、基本的な動作は前述と同様である。

【 0 1 6 1 】

これにより、単純な処理を行うローカルサーバ 3 が複数存在する大規模なネットワーク指向のソフトウェア流通システムにおいても、信頼性を向上させること

ができる。

【0162】

実施の形態11.

なお、上記実施の形態8（図17参照）では、センターサーバ1に接続される全てのローカルサーバ3a～3nに高信頼性手段34a～34nを設けたが、一部のローカルサーバ3のみに高信頼性手段34を設け、他のローカルサーバ3から高信頼性手段34を削除してもよい。

【0163】

以下、図20を参照しながら、複数のうちの一部のローカルサーバ3のみに高信頼性手段34を設けたこの発明の実施の形態11について説明する。

図20はこの発明の実施の形態11を示すブロック構成図であり、前述（図17参照）と同様のものについては、同一符号を付して詳述を省略する。

【0164】

図20において、複数のローカルサーバ3a～3nのうち、一部のローカルサーバ3aは、高信頼性手段34が削除された構成となっており、他のローカルサーバ3nは、高信頼性手段34nを備えている。

【0165】

図20に示したこの発明の実施の形態11による基本的動作は、各ローカルサーバ3a～3nに関して前述と同様である。

【0166】

図20の構成は、高信頼性手段34nが必要なローカルサーバ3nと、高信頼性手段34が不要なローカルサーバ3aとが混在するシステムにおいて、有効である。

【0167】

この場合も、障害対策手段13を備えたセンターサーバ1が論理的に1台あればよく、基本的な動作は前述と同様である。また、システム規模の増大に応じてセンターサーバ1の台数を増やしてもよい。

【0168】

これにより、単純な処理を行うローカルサーバ3aと、通常の高信頼性処理を

行うローカルサーバ 3 n とが複数混在する大規模なネットワーク指向のソフトウェア流通システムにおいても、信頼性を向上させることができる。

【0 1 6 9】

実施の形態 1 2.

なお、上記実施の形態 1 では、センターサーバ 1 にスクリプト 1 2 を設け、ローカルサーバ 3 にスクリプト解釈手段 3 3 を設けたが、スクリプト 1 2 およびスクリプト解釈手段 3 3 を削除してもよい。

【0 1 7 0】

この場合、アプリケーション 1 1 のみで動作可能なシンプルなシステムにおいて有効であり、センターサーバ 1 およびローカルサーバ 3 の構成を簡略化することができる。

【0 1 7 1】

また、障害発生時の障害対策については、前述と同様の作用効果を奏する。

また、センターサーバ 1 に複数のローカルサーバ 3 が接続されたシステムにも適用可能である。

【0 1 7 2】

【発明の効果】

以上のように、この発明の請求項 1 によれば、センターサーバと、広域ネットワークを介してセンターサーバに接続されたローカルサーバとを備えたネットワークシステムからなるソフトウェア管理システムにおいて、センターサーバは、ローカルサーバにダウンロードさせて動作するアプリケーションと、アプリケーションの動作を記述したスクリプトと、障害発生時の対応を行う障害対策手段とを備え、ローカルサーバは、ネットワーク指向言語実行環境と、アプリケーションのセンターサーバからのダウンロードや処理終了後の削除などを行うリモート管理手段と、スクリプトを解釈してアプリケーションに処理を要求するスクリプト解釈手段と、アプリケーションの実行中に発生するイベント情報の記録や、障害発生時の情報管理および復旧処理を行う高信頼性手段とを備え、広域ネットワークを介してダウンロードされたソフトウェアの安全確実な動作を保証するとともに、万一異常が発生しても障害情報収集や復旧を支援する機能を提供したので

、広域ネットワーク上を流通するソフトウェアの処理の信頼性を向上させたソフトウェア管理システムが得られる効果がある。

【 0 1 7 3 】

また、この発明の請求項 2 によれば、請求項 1 において、リモート管理手段は、センターサーバ情報およびアプリケーション情報と、アプリケーションの実行要求に対応する要求処理手段と、センターサーバ情報およびアプリケーション情報に基づいてアプリケーションをセンターサーバからダウンロードするアプリケーションダウンロード手段と、アプリケーション情報に基づいてアプリケーションの起動や削除などの処理を行うアプリケーション管理手段とを含むので、広域ネットワーク上を流通するソフトウェアの処理の信頼性を向上させたソフトウェア管理システムが得られる効果がある。

【 0 1 7 4 】

また、この発明の請求項 3 によれば、請求項 1 において、スクリプト解釈手段は、スクリプト定義およびイベント一覧と、スクリプト定義にしたがってスクリプトを解釈してスクリプトの定義内容に相当するイベントを出力する解釈手段と、イベントを取り込むとともにイベント一覧にしたがってイベントにより駆動される処理を抽出するイベント駆動手段とを含むので、広域ネットワーク上を流通するソフトウェアの処理の信頼性を向上させたソフトウェア管理システムが得られる効果がある。

【 0 1 7 5 】

また、この発明の請求項 4 によれば、請求項 1 において、スクリプトは、XML (eXtensible Markup Language) により記述され、スクリプト定義は、DTD (Document Type Definition) により記述されたので、広域ネットワーク上を流通するソフトウェアの処理の信頼性を向上させたソフトウェア管理システムが得られる効果がある。

【 0 1 7 6 】

また、この発明の請求項 5 によれば、請求項 1 において、障害対策手段は、障害発生時の障害情報をローカルサーバから取得する障害情報取得手段と、障害発生時の障害対策を決定してローカルサーバに通知する障害対策通知手段とを含み

、高信頼性手段は、障害発生時に障害の発生を検知する障害検知手段と、障害発生時の障害情報を収集する障害情報収集手段と、障害情報をセンターサーバに通知する障害通知手段と、センターサーバからの障害対策に応じて障害を復旧させる復旧手段と、イベント情報を収集記録するイベント収集手段とを含むので、広域ネットワーク上を流通するソフトウェアの処理の信頼性を向上させたソフトウェア管理システムが得られる効果がある。

【 0 1 7 7 】

また、この発明の請求項 6 によれば、請求項 5 において、障害対策手段は、障害情報の種別毎に障害対策を格納する障害対策一覧表を含み、障害対策通知手段は、障害情報の種別に対応した障害対策を障害対策一覧表に基づいて検出する障害対策検出手段と、障害対策をローカルサーバに通知する通知手段とを含むので、広域ネットワーク上を流通するソフトウェアの処理の信頼性を向上させたソフトウェア管理システムが得られる効果がある。

【 0 1 7 8 】

また、この発明の請求項 7 によれば、請求項 1 において、障害対策手段は、障害発生時の障害情報をローカルサーバから取得する障害情報取得手段を含み、高信頼性手段は、障害発生時に障害の発生を検知する障害検知手段と、障害発生時の障害情報を収集する障害情報収集手段と、障害発生時の障害への対応を自律的に行い障害を自動復旧させる復旧手段と、障害情報および自動復旧の情報をセンターサーバに通知する通知手段と、イベント情報を収集記録するイベント収集手段とを含むので、広域ネットワーク上を流通するソフトウェアの処理の信頼性を向上させたソフトウェア管理システムが得られる効果がある。

【 0 1 7 9 】

また、この発明の請求項 8 によれば、請求項 1 において、ローカルサーバは複数存在し、それぞれ、ネットワーク指向言語実行環境、リモート管理手段、スクリプト解釈手段および高信頼性手段を備えたので、大規模なシステムに対しても、広域ネットワーク上を流通するソフトウェアの処理の信頼性を向上させたソフトウェア管理システムが得られる効果がある。

【 0 1 8 0 】

また、この発明の請求項 9 によれば、請求項 8 において、センターサーバは複数存在し、それぞれ、アプリケーション、スクリプトおよび障害対策手段を備えたので、大規模なシステムに対しても、センターサーバの負荷を軽減させるとともに冗長性をもたせ、広域ネットワーク上を流通するソフトウェアの処理の信頼性を向上させたソフトウェア管理システムが得られる効果がある。

【 0 1 8 1 】

また、この発明の請求項 1 0 によれば、請求項 1 において、ローカルサーバは複数存在し、少なくとも 1 つのローカルサーバは、ネットワーク指向言語実行環境、リモート管理手段、スクリプト解釈手段および高信頼性手段を備え、他のローカルサーバは、ネットワーク指向言語実行環境、リモート管理手段およびスクリプト解釈手段を備えたので、大規模なシステムに対しても、広域ネットワーク上を流通するソフトウェアの処理の信頼性を向上させたソフトウェア管理システムが得られる効果がある。

【 0 1 8 2 】

また、この発明の請求項 1 1 によれば、センターサーバと、広域ネットワークを介してセンターサーバに接続されたローカルサーバとを備えたネットワークシステムからなるソフトウェア管理システムにおいて、センターサーバは、ローカルサーバにダウンロードさせて動作するアプリケーションと、アプリケーションの動作を記述したスクリプトとを備え、ローカルサーバは、ネットワーク指向言語実行環境と、アプリケーションのセンターサーバからのダウンロードや処理終了後の削除などを行うリモート管理手段と、スクリプトを解釈してアプリケーションに処理を要求するスクリプト解釈手段とを備えたので、ローカルサーバの機能構成を簡略化するとともに、広域ネットワーク上を流通するソフトウェアの処理の信頼性を向上させたソフトウェア管理システムが得られる効果がある。

【 0 1 8 3 】

また、この発明の請求項 1 2 によれば、請求項 1 1 において、ローカルサーバは複数存在し、それぞれ、ネットワーク指向言語実行環境、リモート管理手段およびスクリプト解釈手段を備えたので、大規模なシステムに対しても、ローカルサーバの機能構成を簡略化するとともに、広域ネットワーク上を流通するソフト

ウェアの処理の信頼性を向上させたソフトウェア管理システムが得られる効果がある。

【0184】

また、この発明の請求項13によれば、センターサーバと、広域ネットワークを介してセンターサーバに接続されたローカルサーバとを備えたネットワークシステムからなるソフトウェア管理システムにおいて、センターサーバは、ローカルサーバにダウンロードさせて動作するアプリケーションと、障害発生時の対応を行う障害対策手段とを備え、ローカルサーバは、ネットワーク指向言語実行環境と、アプリケーションのセンターサーバからのダウンロードや処理終了後の削除などを行うリモート管理手段と、アプリケーションの実行中に発生するイベント情報の記録や、障害発生時の情報管理および復旧処理を行う高信頼性手段とを備えたので、センターサーバの機能構成を簡略化するとともに、広域ネットワーク上を流通するソフトウェアの処理の信頼性を向上させたソフトウェア管理システムが得られる効果がある。

【0185】

また、この発明の請求項14によれば、請求項13において、ローカルサーバは複数存在し、それぞれ、ネットワーク指向言語実行環境、リモート管理手段および高信頼性手段を備えたので、大規模なシステムに対しても、センターサーバの機能構成を簡略化するとともに、広域ネットワーク上を流通するソフトウェアの処理の信頼性を向上させたソフトウェア管理システムが得られる効果がある。

【図面の簡単な説明】

【図1】 この発明の実施の形態1を示すブロック構成図である。

【図2】 この発明の実施の形態1による動作を示すフローチャートである。

【図3】 この発明の実施の形態2を示すブロック構成図である。

【図4】 この発明の実施の形態2による動作を示すフローチャートである。

【図5】 この発明の実施の形態3の要部を示すブロック構成図である。

【図6】 この発明の実施の形態3によるスクリプト定義の一例を示す説明

図である。

【図 7】 この発明の実施の形態 3 によるイベント一覧の一例を示す説明図である。

【図 8】 この発明の実施の形態 3 による動作を示すフローチャートである。

【図 9】 この発明の実施の形態 4 によるサービス記述例を示す説明図である。

【図 1 0】 この発明の実施の形態 4 によるサービス記述例を示す説明図である。

【図 1 1】 この発明の実施の形態 5 を示すブロック構成図である。

【図 1 2】 この発明の実施の形態 5 による動作を示すフローチャートである。

【図 1 3】 この発明の実施の形態 6 を示すブロック構成図である。

【図 1 4】 この発明の実施の形態 6 による動作を示すフローチャートである。

【図 1 5】 この発明の実施の形態 7 を示すブロック構成図である。

【図 1 6】 この発明の実施の形態 7 による動作を示すフローチャートである。

【図 1 7】 この発明の実施の形態 8 を示すブロック構成図である。

【図 1 8】 この発明の実施の形態 9 を示すブロック構成図である。

【図 1 9】 この発明の実施の形態 1 0 を示すブロック構成図である。

【図 2 0】 この発明の実施の形態 1 1 を示すブロック構成図である。

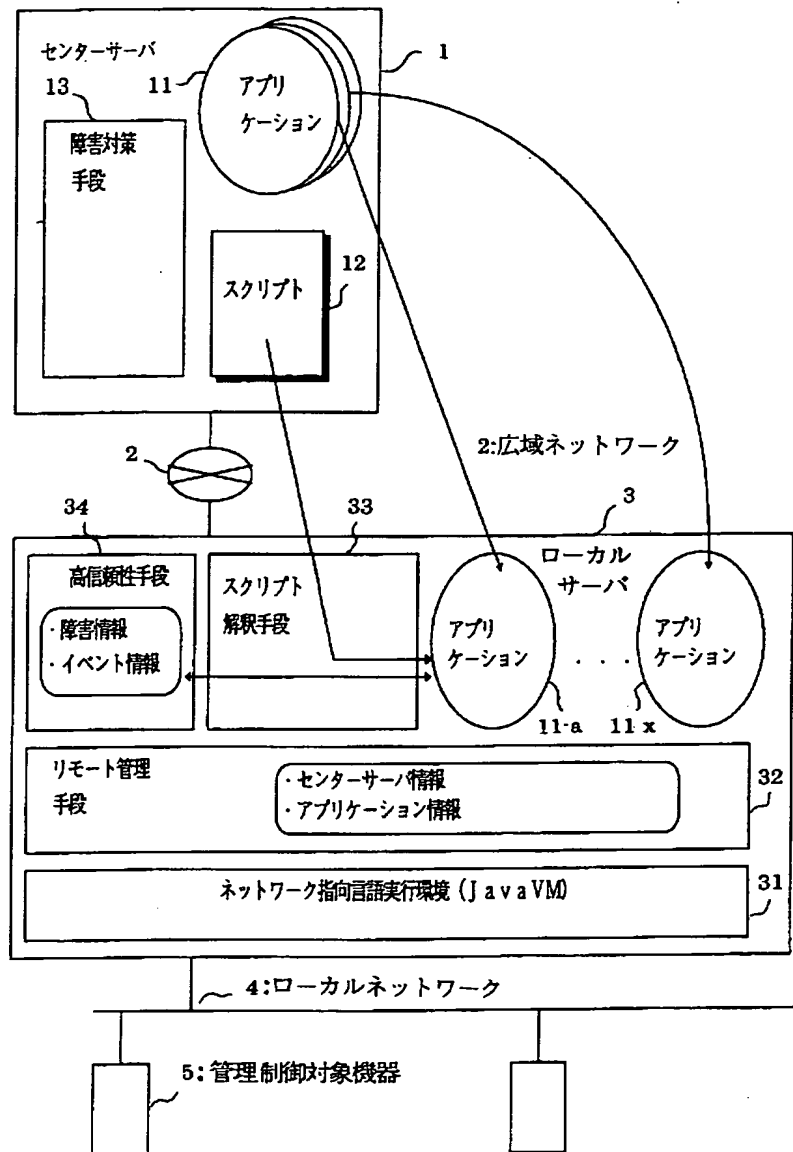
【符号の説明】

1 センターサーバ、2 広域ネットワーク、3、3 a ～ 3 n ローカルサーバ、4、4 a ～ 4 n ローカルネットワーク、5、5 a ～ 5 n 管理制御対象機器、1 1、1 1 a ～ 1 1 x アプリケーション、1 2、1 2 a スクリプト、1 3 障害対策手段、3 1、3 1 a ～ 3 1 n ネットワーク指向言語実行環境（Java VM）、3 2、3 2 a ～ 3 2 n リモート管理手段、3 3、3 3 a ～ 3 3 n スクリプト解釈手段、3 4、3 4 a ～ 3 4 n 高信頼性手段、1 3 1 障

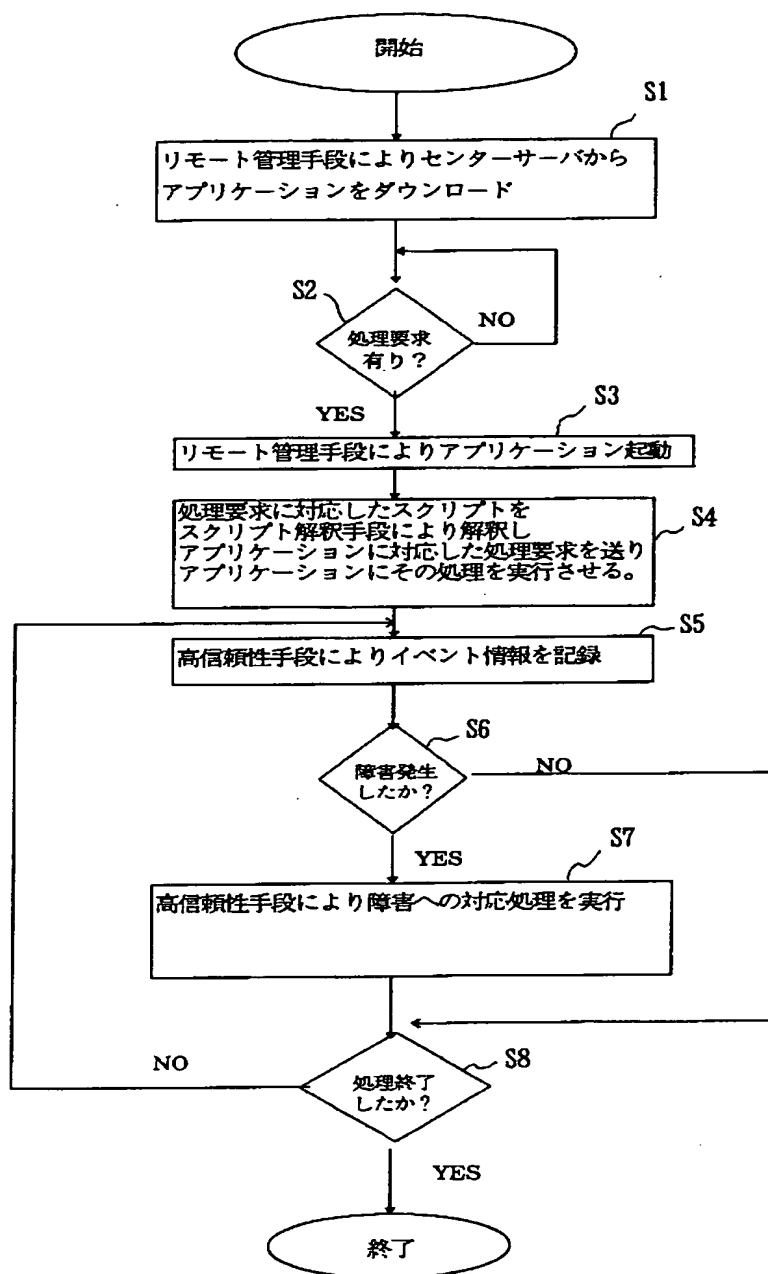
害情報取得手段、132 障害対策通知手段、133 障害対策一覧表、321
センターサーバ情報、322 アプリケーション情報、323 要求処理手段
、324 アプリケーションダウンロード手段、325 アプリケーション管理
手段、331 スクリプト定義、332 イベント一覧、333 解釈手段、
334 イベント駆動手段、341 障害情報、342 イベント情報、343
障害検知手段、344 障害情報収集手段、345 障害通知手段、346、
348 復旧手段、347 イベント収集手段、349 通知手段、1321
障害対策検出手段、1322 通知手段。

【書類名】 図面

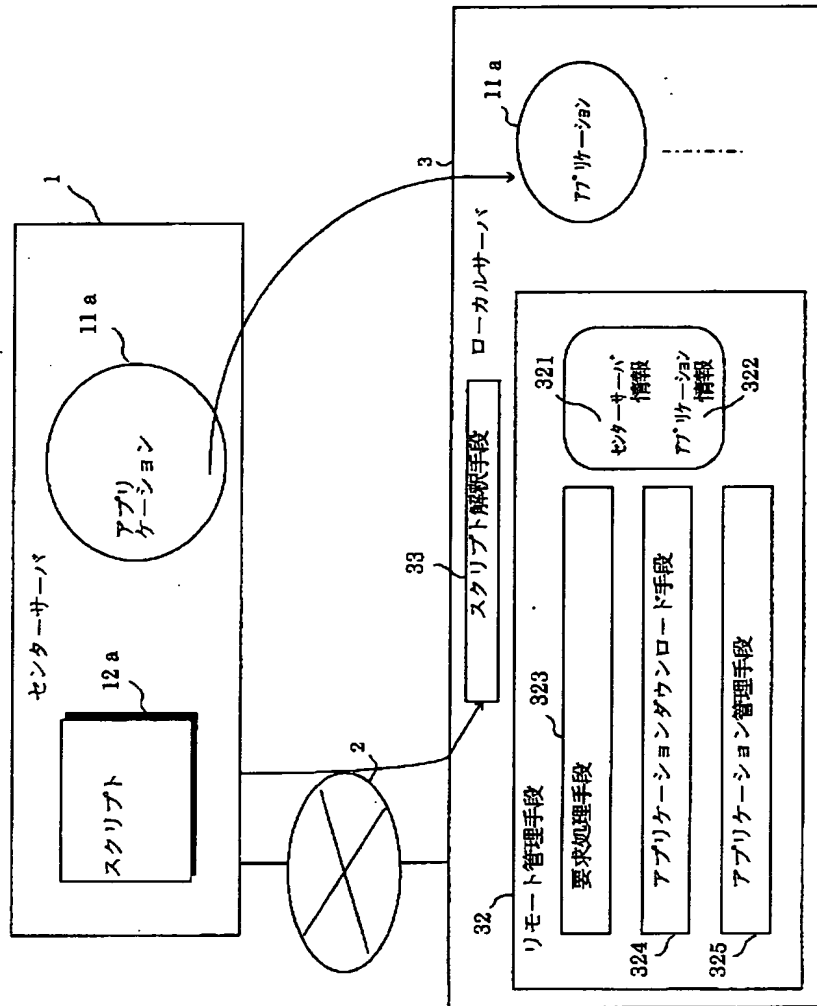
【図 1】



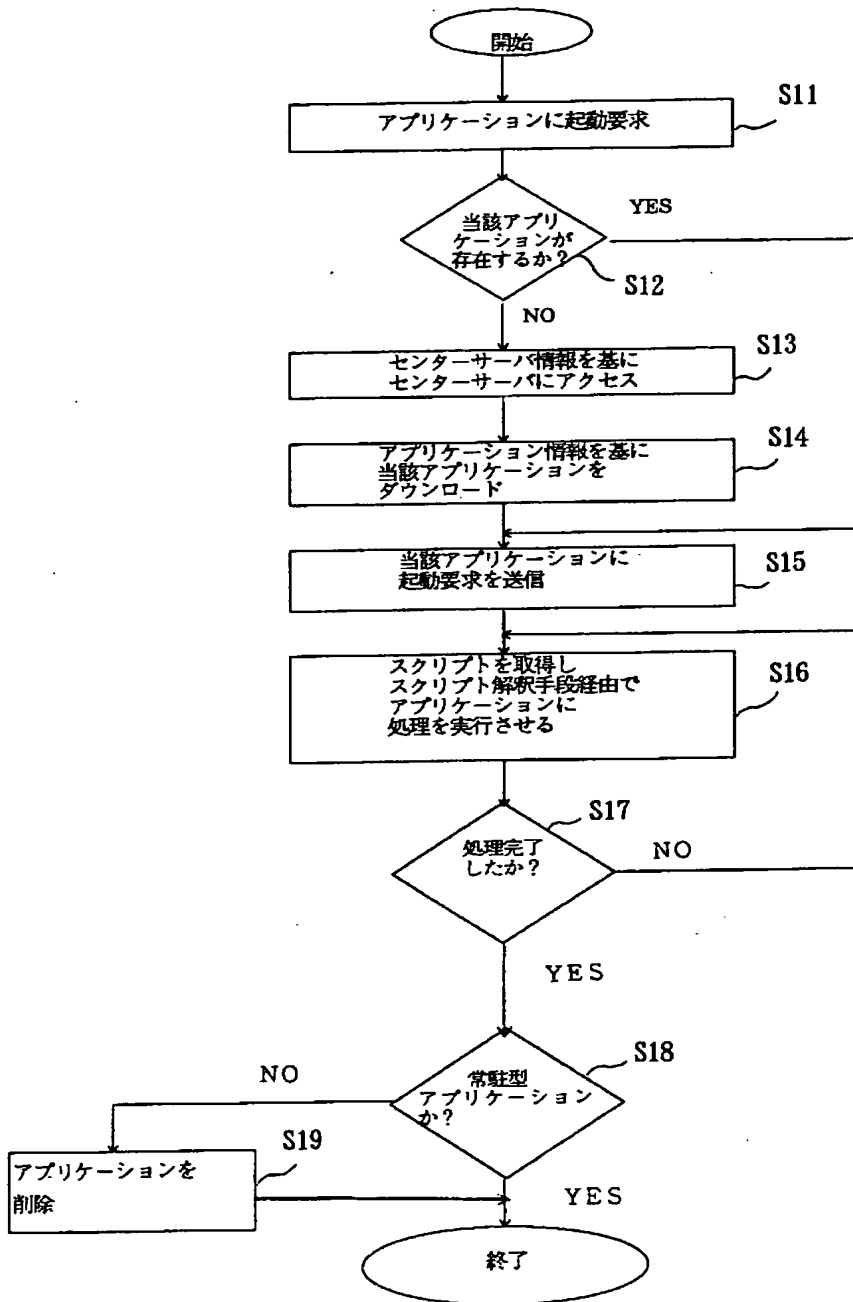
【図 2】



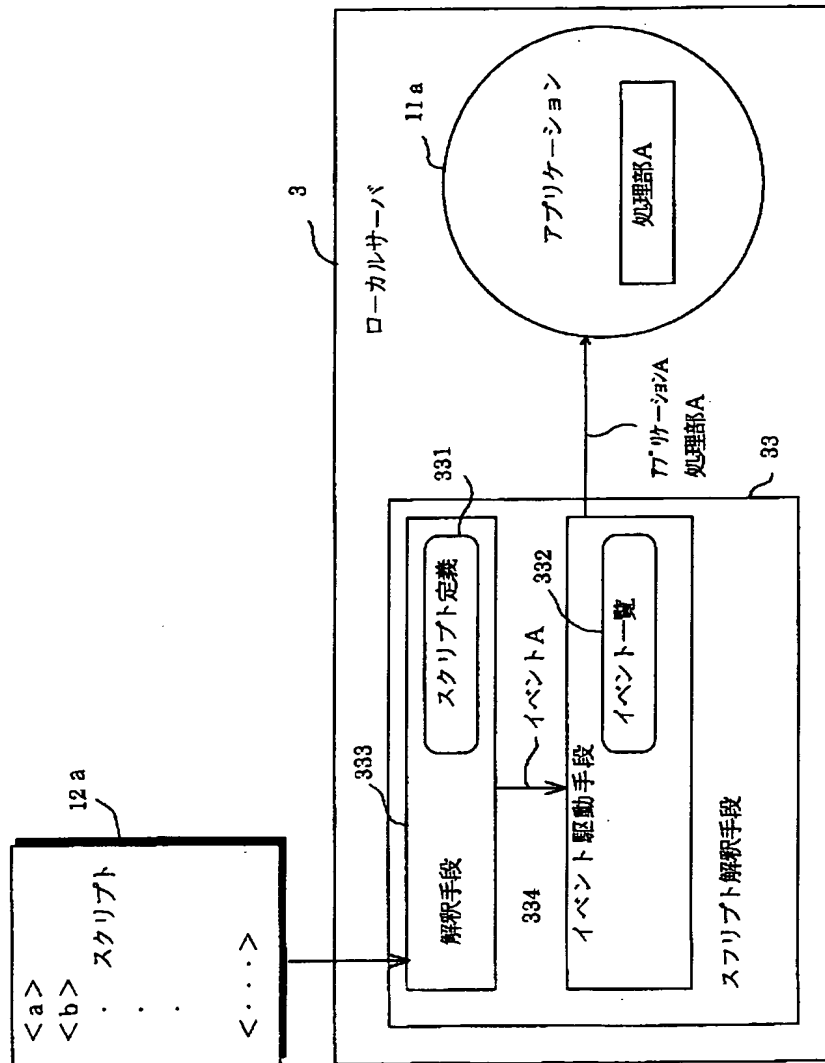
【図 3】



【図4】



【図 5】



【図 6】

331

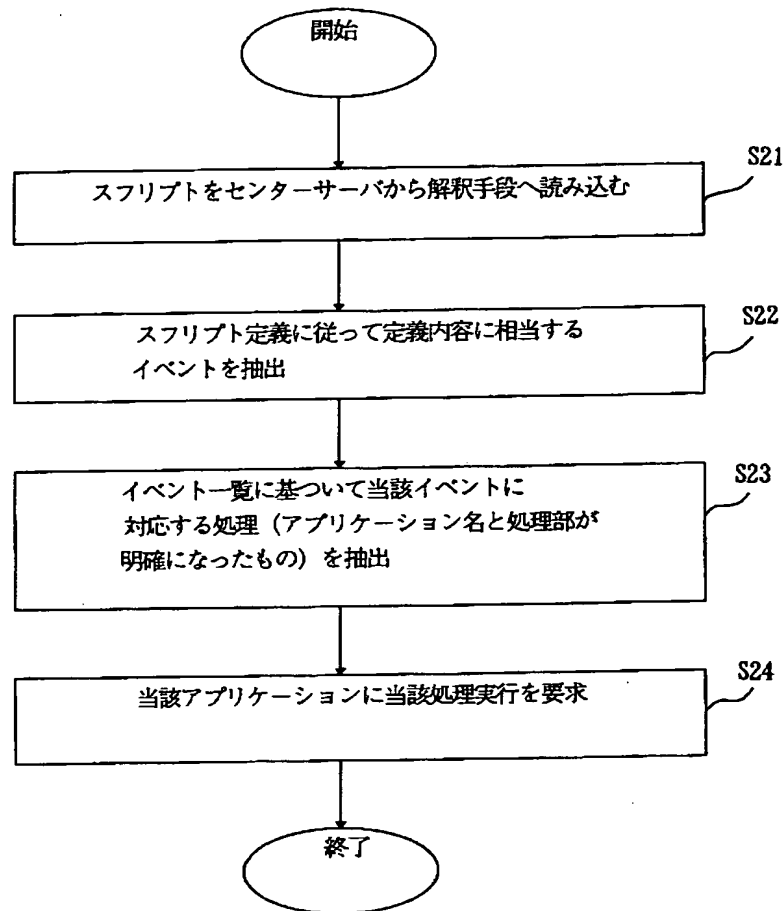
スクリプト定義	イベント
⋮	
<スクリプト a>	イベント A
⋮	

【図 7】

332

イベント	処理
⋮	⋮
イベント A	アプリケーション 11-a 処理部 A
⋮	⋮

【図 8】



【図 9】

```

<? XML version="1.0" ?>
<エアコン制御>
  <初期化>
    <エアコン>
      <温度 単位="摂氏">28<?温度>
      <風速>静</風速>
    </エアコン>
    <温度センサ>
      .
    </温度センサ>
  </初期化>
  <自動制御>
    .
  </自動制御>
</エアコン制御>

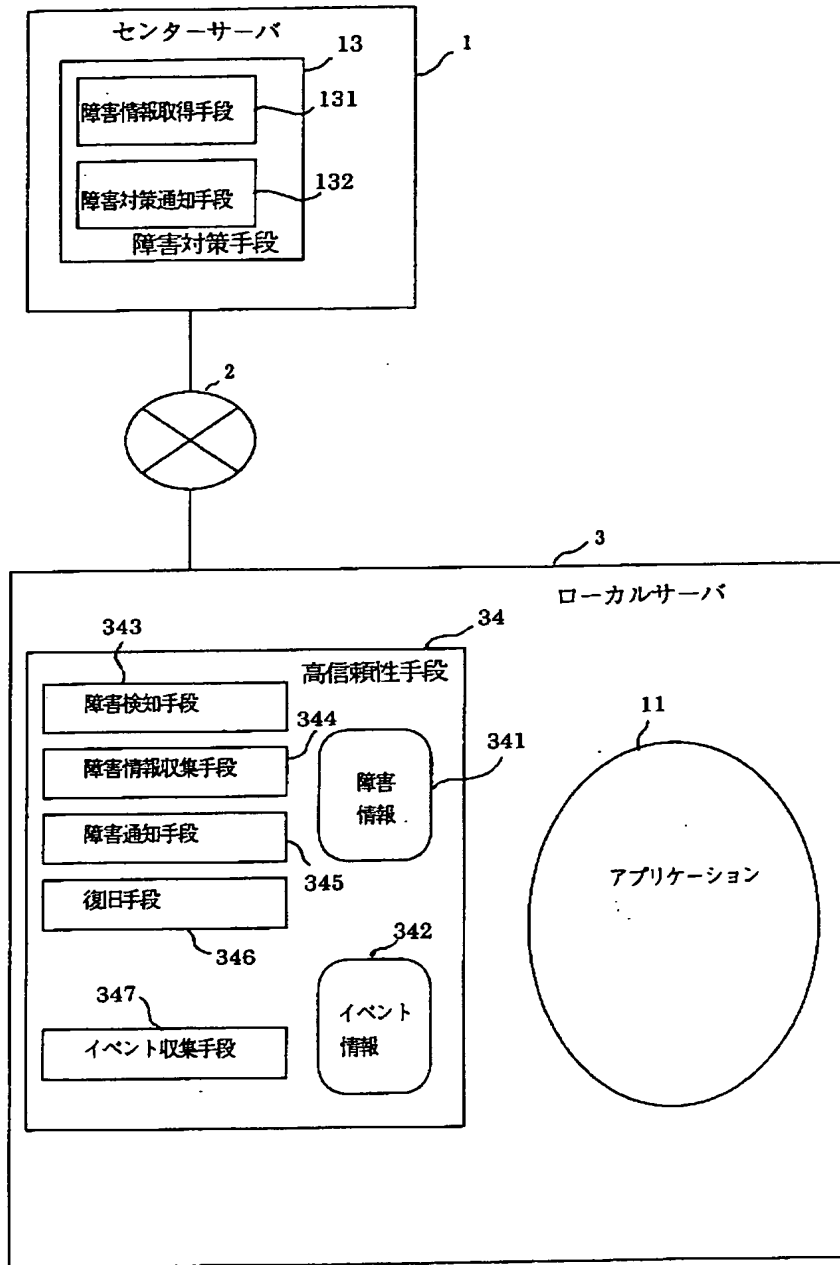
```

【図 1 0】

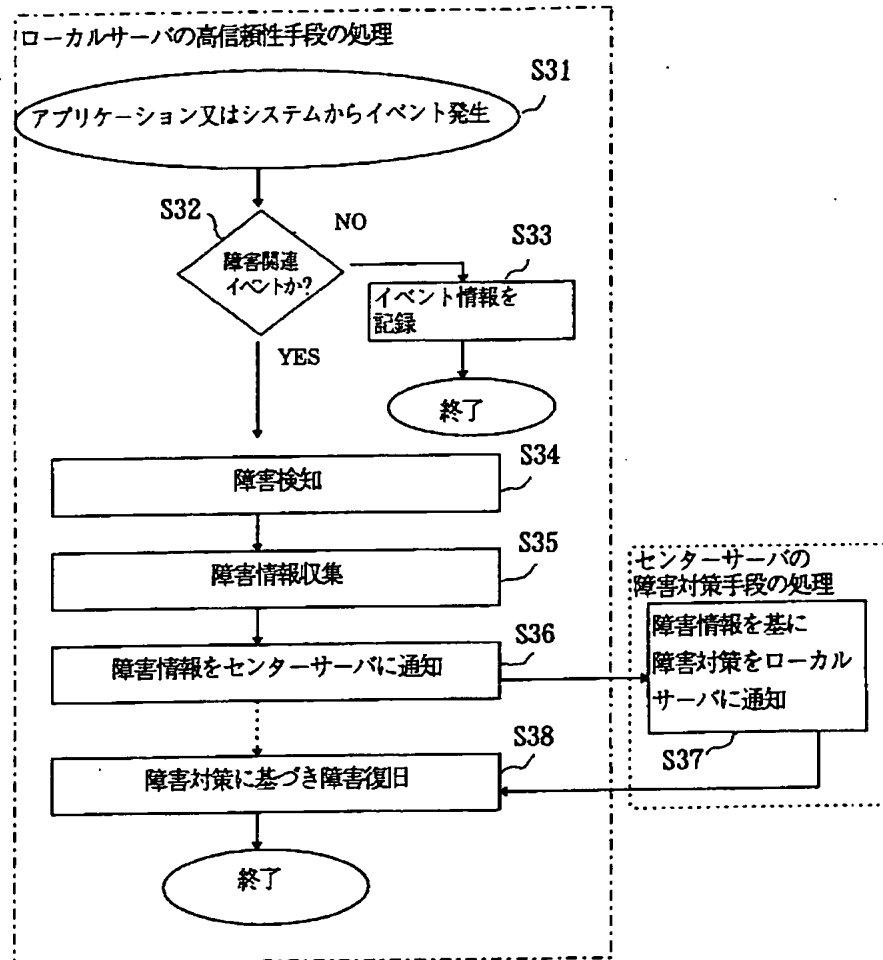
XML 文書に対する DTD

```
<!DOCTYPE エアコン制御 [  
<!ELEMENT エアコン制御 (初期化, 自動制御) >  
<!ELEMENT 初期化 (エアコン, 温度センサ) >  
<!ELEMENT 自動制御 (...) >  
<!ELEMENT エアコン (温度, 風速) >  
<!ELEMENT センサ (...) >  
<!ELEMENT 温度 (#PCDATA) >  
<!ELEMENT 風速 (#PCDATA) >  
;  
<!ATTRLIST 温度 単位 (摂氏 | 華氏) "摂氏" >
```

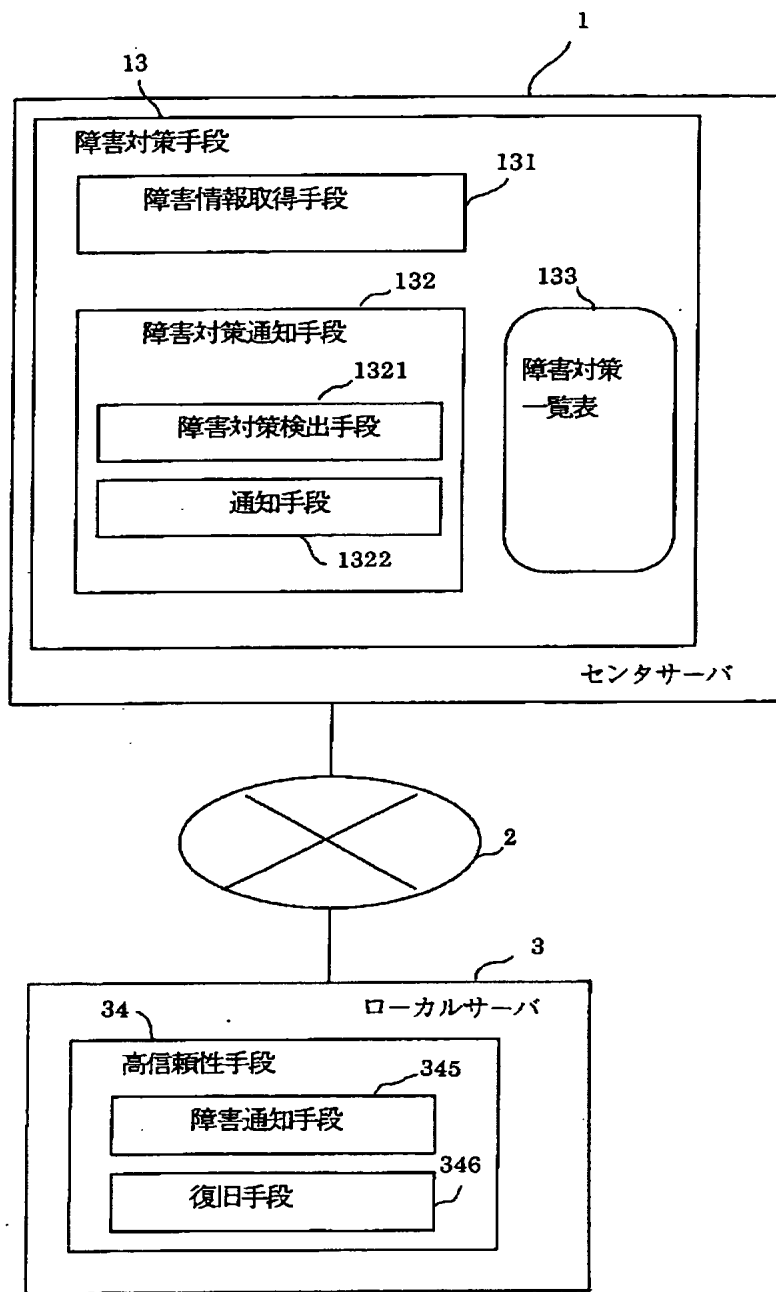
【図 1 1】



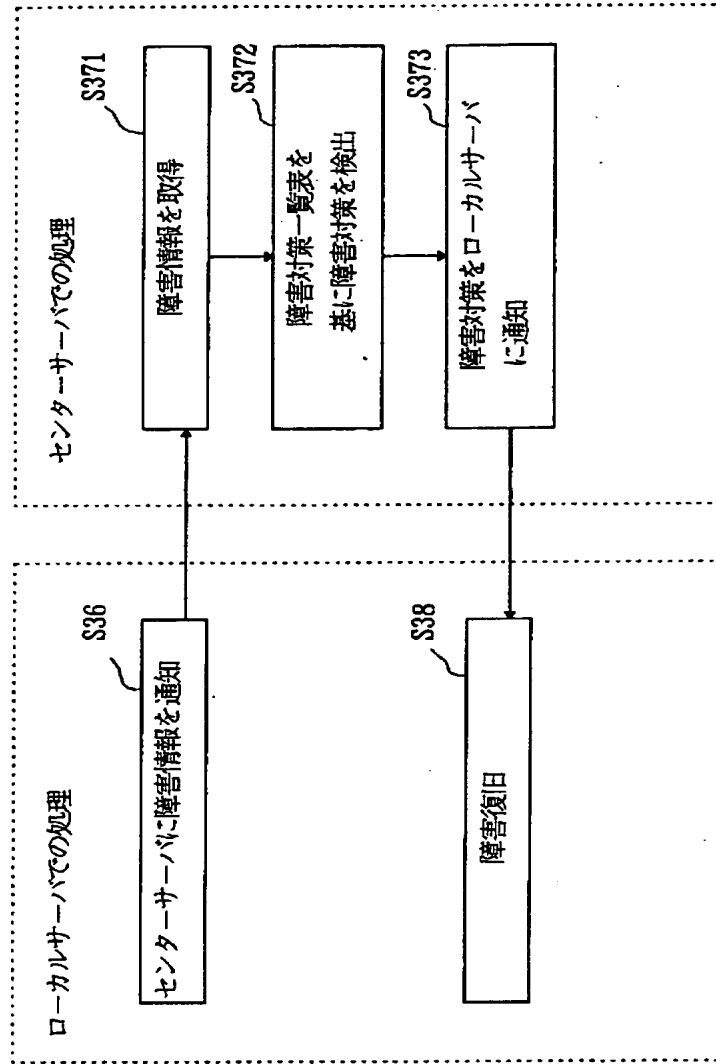
【図 1 2】



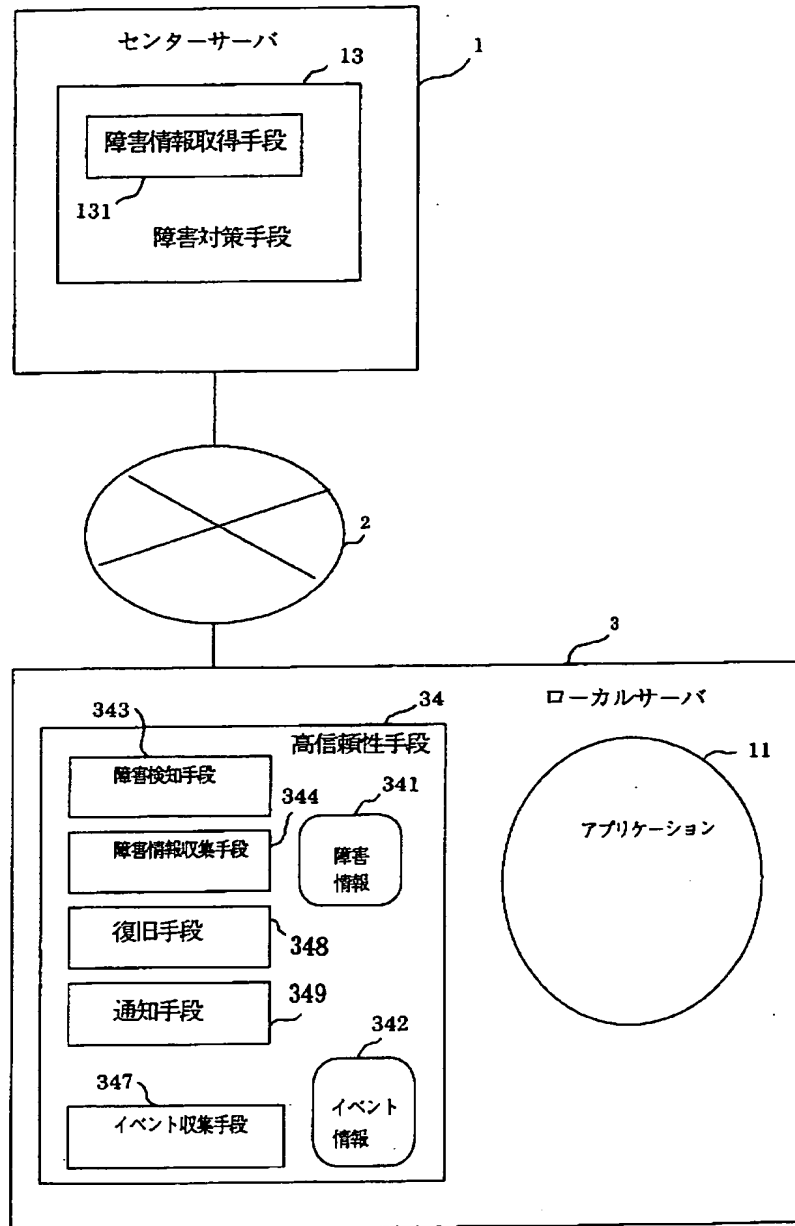
【図 13】



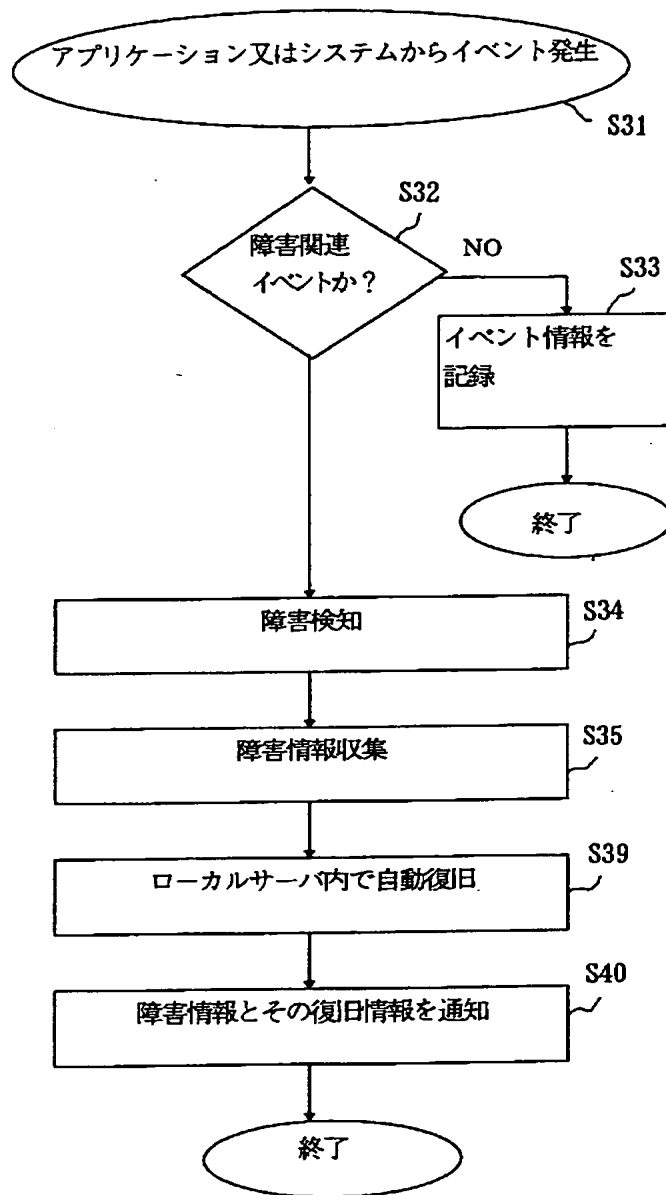
【図 1 4】



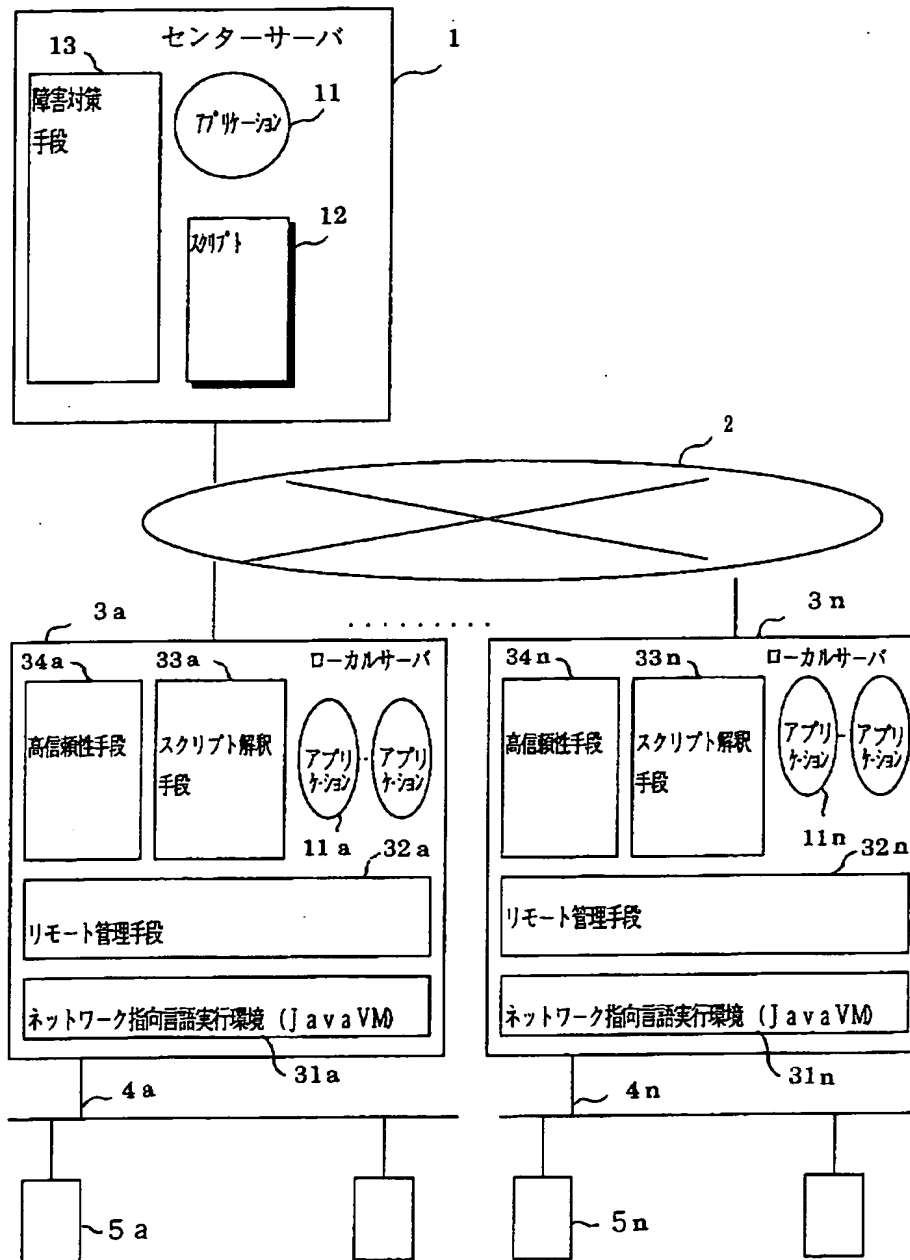
【図 1 5】



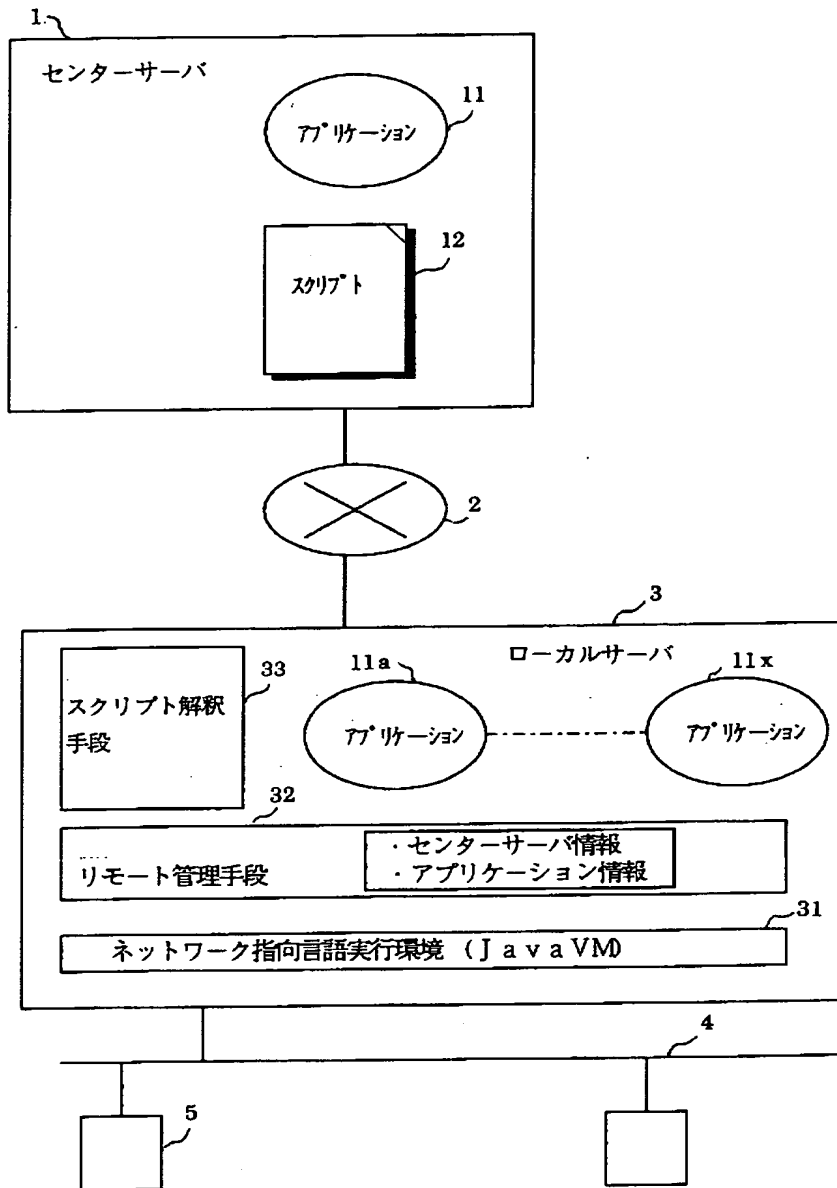
【図 16】



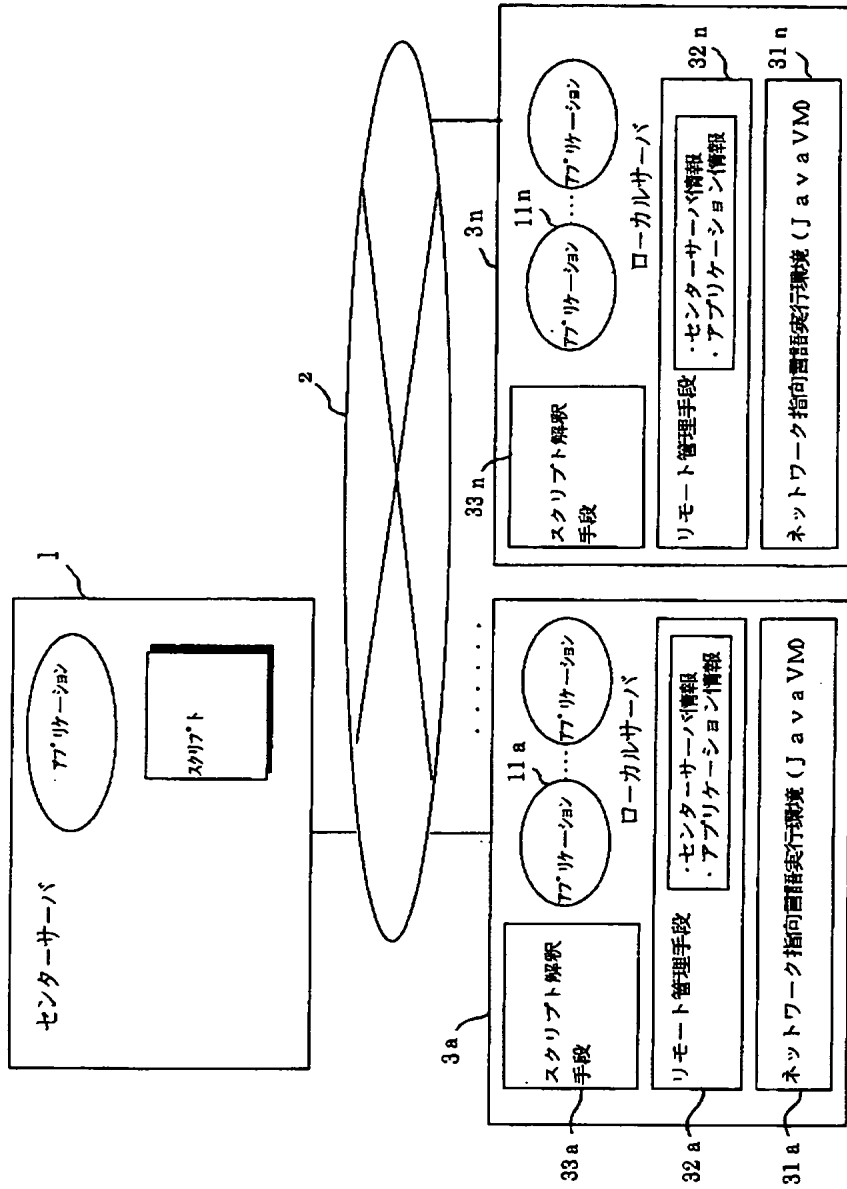
【図 17】



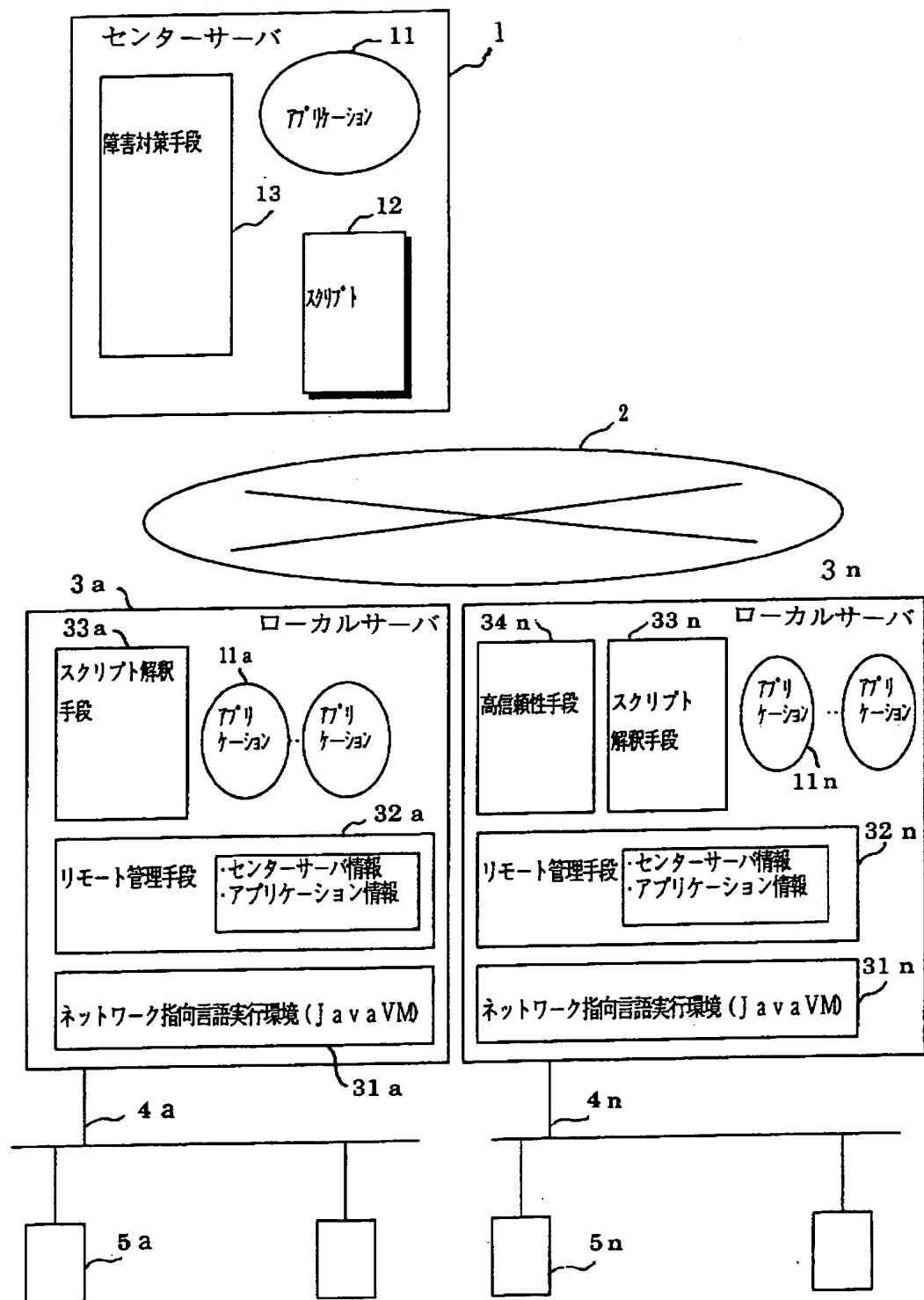
【図 1 8】



【図 19】



【図20】



【書類名】 要約書

【要約】

【課題】 広域ネットワーク上を流通するソフトウェアの処理の信頼性を向上させたソフトウェア管理システムを得る。

【解決手段】 センターサーバ 1 は、アプリケーション 1 1 の動作を記述したスクリプト 1 2 と、障害発生時の対応を行う障害対策手段 1 3 とを備え、ローカルサーバ 3 は、ネットワーク指向言語実行環境 3 1 と、アプリケーションのダウンロードや削除を行うリモート管理手段 3 2 と、スクリプト解釈手段 3 3 と、アプリケーションの実行中に発生するイベント情報の記録や、障害発生時の情報管理および復旧処理を行う高信頼性手段 3 4 とを備え、広域ネットワーク 2 を介してダウンロードされたソフトウェアの安全確実な動作を保証するとともに、万一異常が発生しても障害情報収集や復旧を支援する機能を提供した。

【選択図】 図 1

出 願 人 履 歴 情 報

識別番号 [000006013]

1. 変更年月日 1990年 8月24日
[変更理由] 新規登録
住 所 東京都千代田区丸の内2丁目2番3号
氏 名 三菱電機株式会社